Chương 6: Tầng ứng dụng

Nội dung

- Vai trò của tầng ứng dụng.
- Mô tả về tầng ứng dụng trong mô hình OSI và TCP/IP.
- Mô hình quản lý các dịch vụ trong tầng ứng dụng.
- Giới thiệu về các dịch vụ và giao thức thông dụng trong tầng ứng dụng.

1. Vai trò của tầng ứng dụng (Application)

- Application cung cấp giao diện chính để người dùng tương tác với chương trình ứng dụng.
- Thông qua các phần mềm, dịch vụ, giao thức sẽ giúp cho người dùng truy nhập các thông tin và dữ liệu trên mạng.
- Một số ví dụ về các ứng dụng trong tầng này bao gồm: Web, Mail, FTP, Telnet...

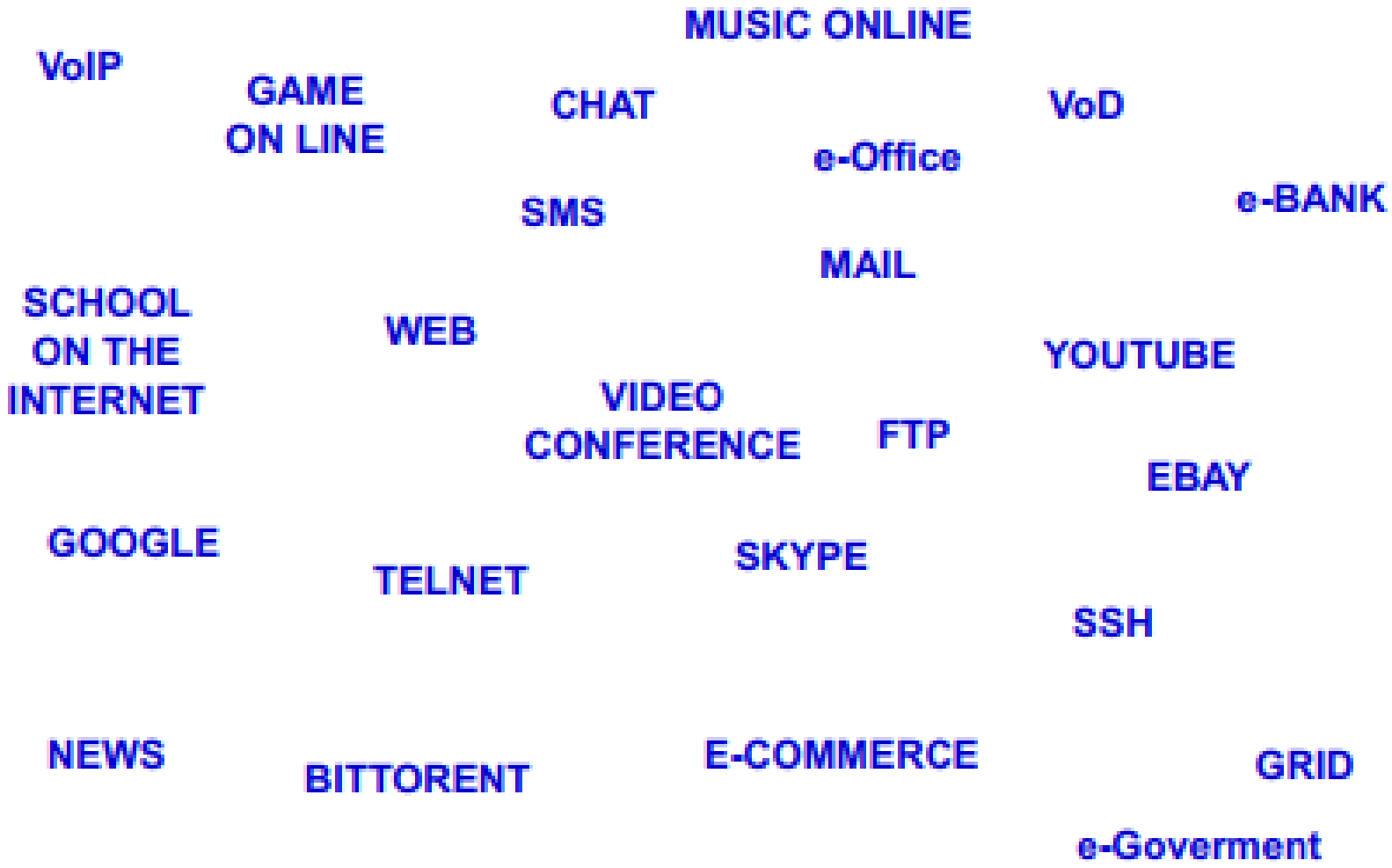
2. Mô tả về tầng ứng dụng trong OSI và TCP/IP

- Trong mô hình OSI thì tầng ứng dụng được tách biệt ra thành Application, Presentation, Session để dễ học và phân tích.

- Tầng ứng dụng được mô tả là tầng để người dùng có thể ra lệnh, yêu cầu các thao tác thông qua cửa sổ của các chương trình ứng dụng. Từ các chương trình ứng dụng ở tầng 7 nó sẽ được Presentation phiên dịch ra để máy tính hiểu, còn tầng Session sẽ đàm phán phiên truyền.

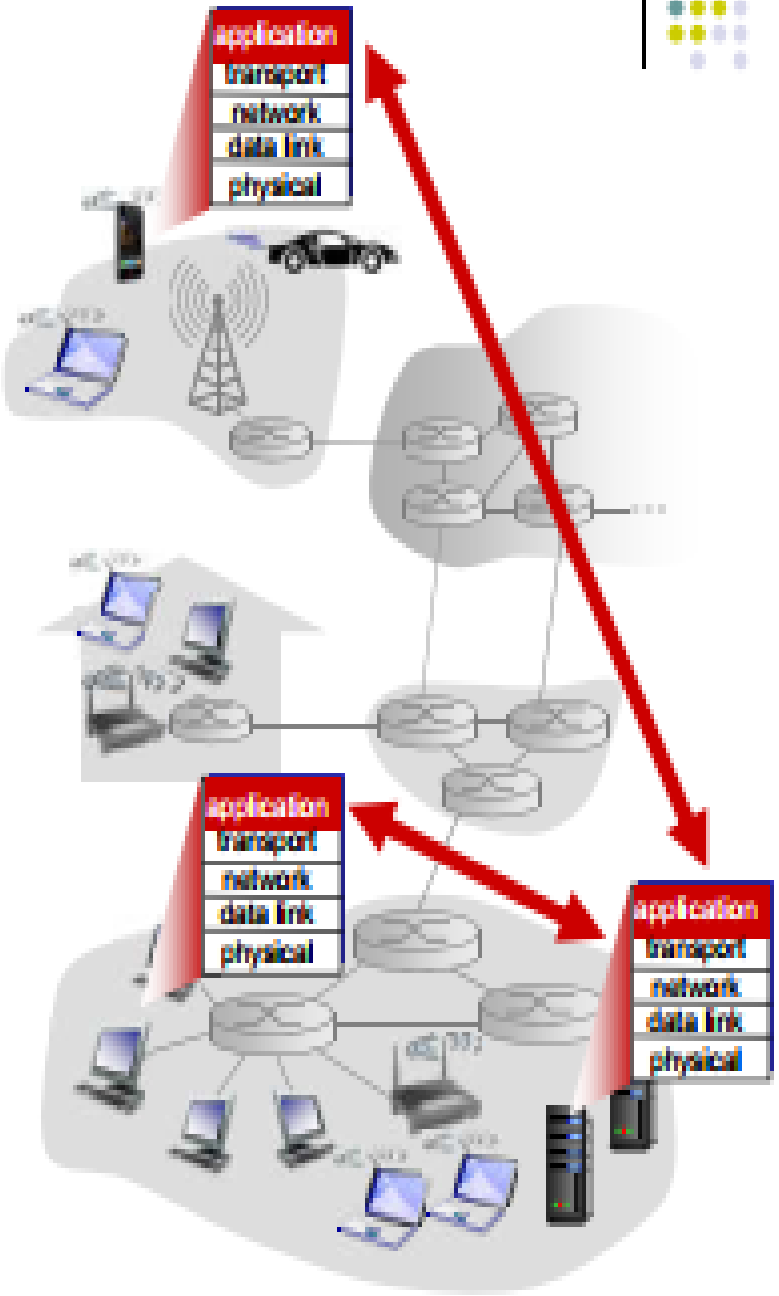
- Trong TCP/IP: Application nó bao gồm cả 3 tầng (đã tham chiếu trong OSI) đó là Application, Presentation, Session

Ứng dụng và dịch vụ trên mạng?



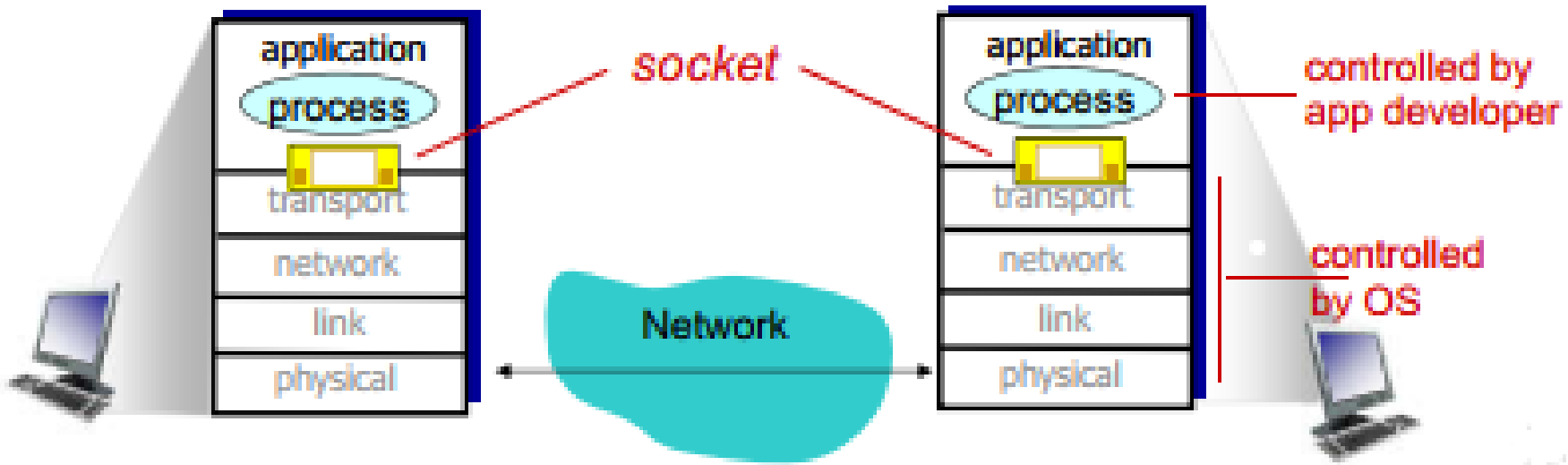
Ứng dụng mạng

- Hoạt động trên các hệ thống đầu cuối (end system)
- Cài đặt giao thức ứng dụng để cung cấp dịch vụ
- Gồm có 2 tiến trình giao tiếp với nhau qua môi trường mạng:
  - Client: cung cấp giao diện NSD, gửi thông điệp yêu cầu dịch vụ/
  - Server: cung cấp dịch vụ, trả thông điệp đáp ứng
- Ví dụ: Web
  - Web browser (trình duyệt Web): Chrome, Firefox…
  - Web server: Apache, Tomcat…



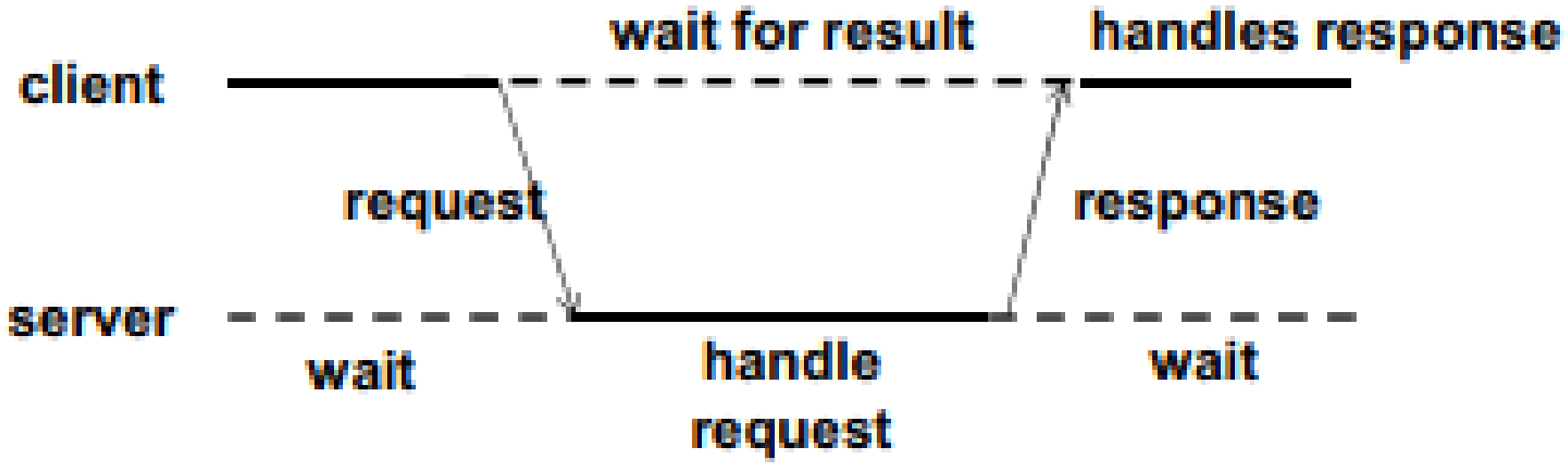
Giao tiếp giữa các tiến trình ứng dụng

- Socket: điểm truy cập dịch vụ của tầng giao vận
- Các tiến trình ứng dụng sử dụng socket gọi dịch vụ của tầng giao vận để trao đổi thông điệp
- Định danh cho tiến trình bởi: Địa chỉ IP, Số hiệu cổng
- Ví dụ: tiến trình web server trên máy chủ của SoICT có định danh 202.191.56.65:80



Giao tiếp giữa các tiến trình

- Tiến trình client: gửi yêu cầu
- Tiến trình server: trả lời
- Mô hình điển hình: 1 server – nhiều client
- Client cần biết địa chỉ của server: địa chỉ IP, số hiệu cổng



3. Mô hình quản lý các dịch vụ trong tầng ứng dụng

•Client/Server: là máy con (đóng vài trò máy khách) gửi một yêu cầu (request) đến máy chủ (đóng vai trò người cung cấp dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy con. Ta thường thấy là dịch Web, Mail, DNS....

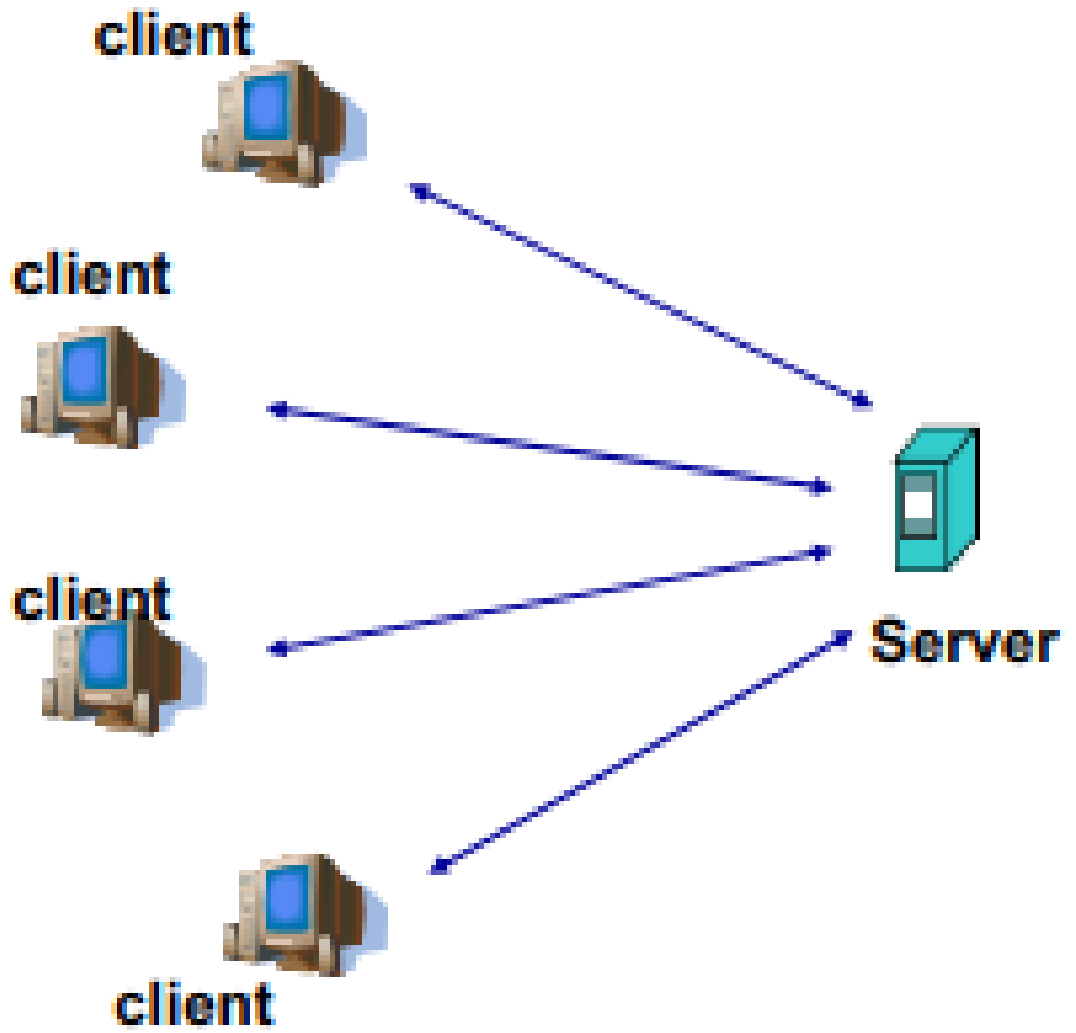
•Peer to Peer: là mạng ngang hàng. Ngang hàng về mặt chức năng và ứng dụng. Hoạt động các ứng dụng đều đặt ở tầng 7. Hai máy trực tiếp nói chuyện với nhau vì tầng 7 không phụ thuộc vào đường truyền.

- Mô hình lai

Các mô hình ứng dụng

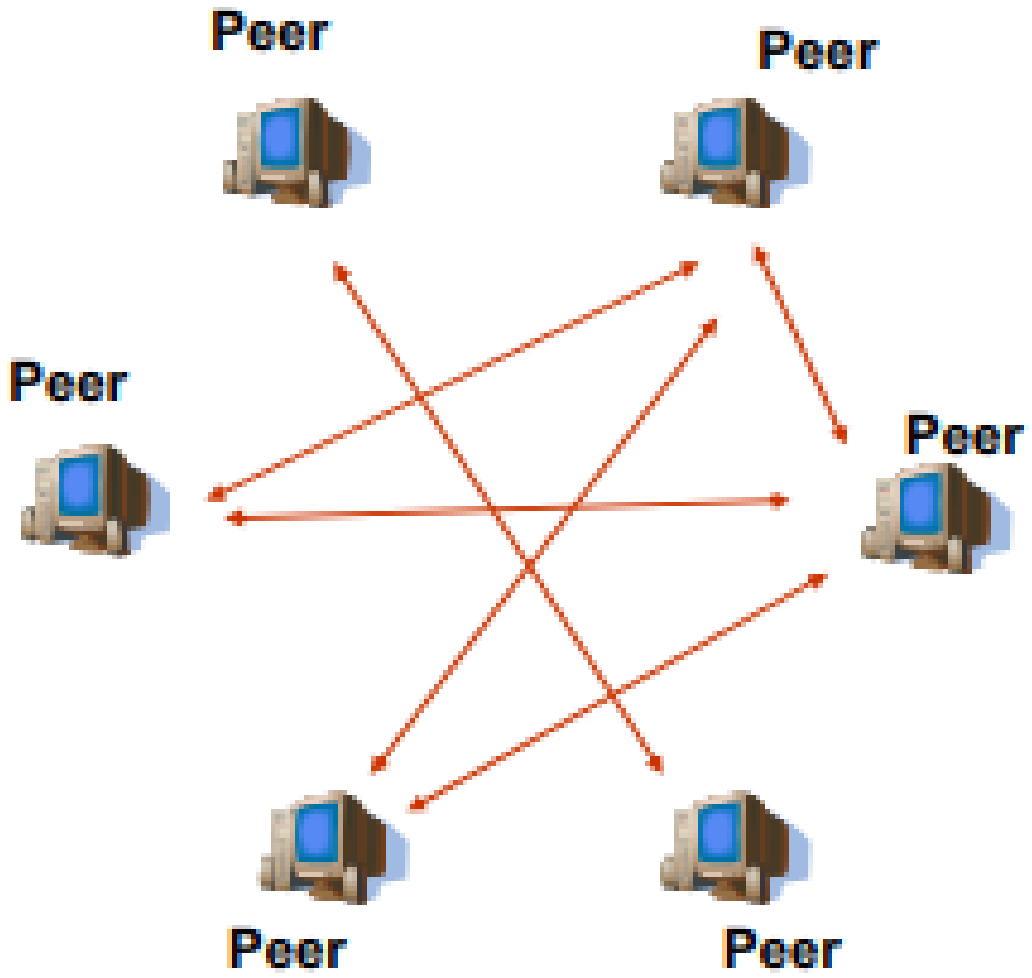
- Khách-chủ (Client/Server)
- Ngang hàng (P2P: Peer-to-peer)

- Khách
  - Gửi yêu cầu truy cập dịch vụ đến máy chủ
  - Về nguyên tắc, không liên lạc trực tiếp với các máy khách khác
- Chủ
  - Thường xuyên online để chờ y/c đến từ máy trạm
  - Có thể có máy chủ dự phòng để nâng cao hiệu năng, phòng sự cố
- e.g. Web, Mail, …

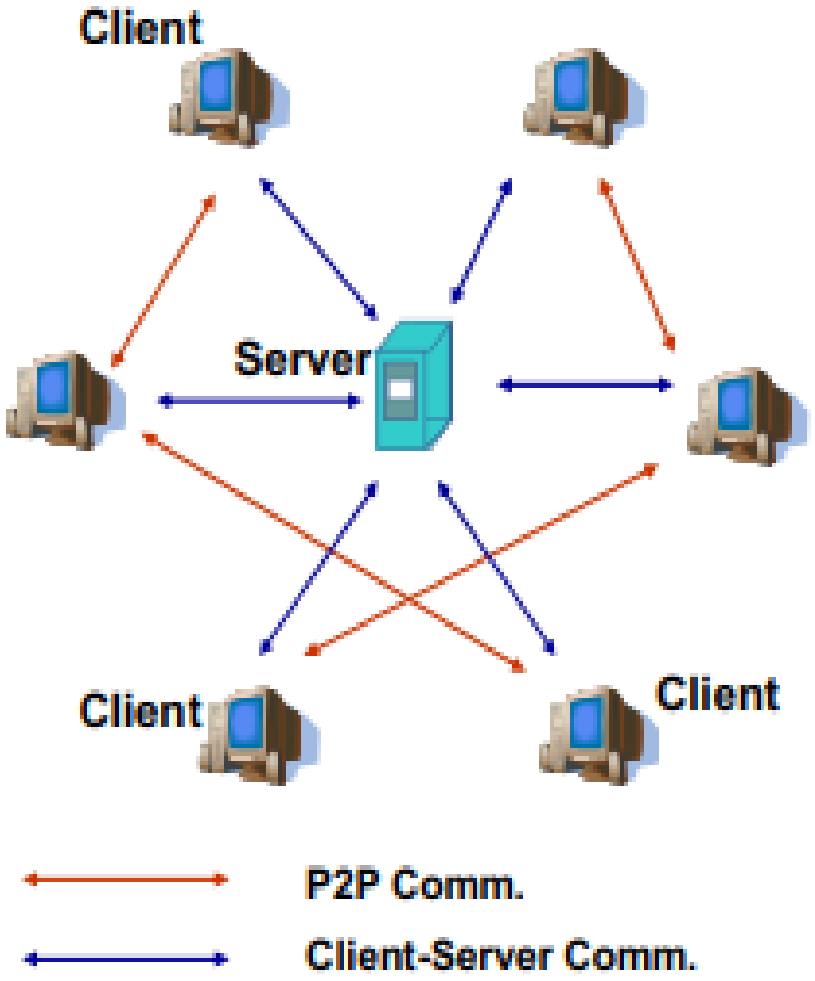


Mô hình ngang hàng thuần túy

- Không có máy chủ trung tâm
- Các máy có vai trò ngang nhau
- Hai máy bất kỳ có thể liên lạc trực tiếp với nhau
- Không cần vào mạng thường xuyên
- E.g. Gnutella



Mô hình lai



- Một máy chủ trung tâm để quản lý NSD, thông tin tìm kiếm…
- Các máy khách sẽ giao tiếp trực tiếp với nhau sau khi đăng nhập
- E.g. Skype
  - Máy chủ Skype quản lý các phiên đăng nhập, mật khẩu…
  - Sau khi kết nối, các máy sẽ gọi VoIP trực tiếp cho nhau

4. Giới thiệu về dịch vụ và giao thức trong tầng ứng dụng

Tên dịch vụ	Tên tiếng anh	Giao thức dịch vụ	Giao thức truyền thông	Port dịch vụ
DNS	Domain Name System	DNS	TCP/UDP	53
Web	+ Hypertext Transfer Protocol + Hypertext Transfer Protocol Secure	HTTP HTTPS	TCP TCP	80 443
Mail	+ Simple Mail Transfer Protocol + Post Office Protocol	SMTP POP (POP3)/IMAP	TCP UDP	25 110, 143
DHCP	+ Dynamic Host Configuration Protocol	DHCP Server DHCP Client	UDP UDP	67 68
FTP	File Transfer Protocol	FTP	TCP	20 (truyền dữ liệu) 21 (truyền thông tin điều khiển)
Telnet	Telnet	Telnet	TCP	23
SSH	Secure Shell	Secure Shell	TCP	22
Chat	Internet chat	IM	TCP	531

Tên miền và dịch vụ DNS

Giới thiệu chung

- DNS (Domain Name System) là Hệ thống phân giải tên được phát minh vào năm 1984 cho Internet, chỉ một hệ thống cho phép thiết lập tương ứng giữa địa chỉ IP và tên miền.

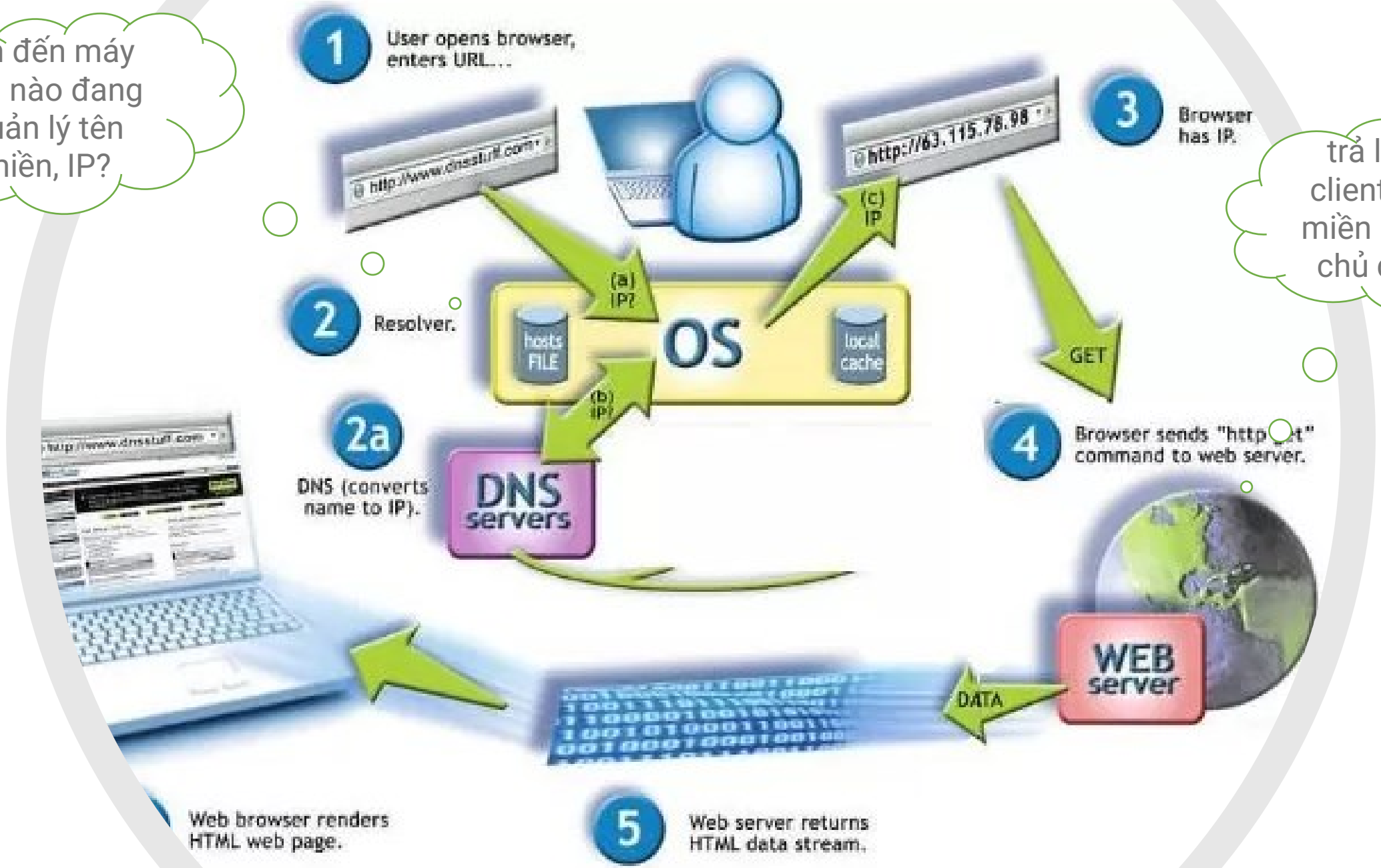
- Hệ thống tên miền (DNS) là một hệ thống đặt tên theo thứ tự cho máy vi tính, dịch vụ, hoặc bất kỳ nguồn lực tham gia vào Internet.

- Liên kết nhiều thông tin đa dạng với tên miền được gán cho những người tham gia. Quan trọng nhất là, chuyển tên miền có ý nghĩa cho con người vào số định danh (nhị phân), liên kết với các trang thiết bị mạng cho các mục đích định vị và địa chỉ hóa các thiết bị khắp thế giới.

- Phân giải tên máy chủ máy tính thành địa chỉ IP

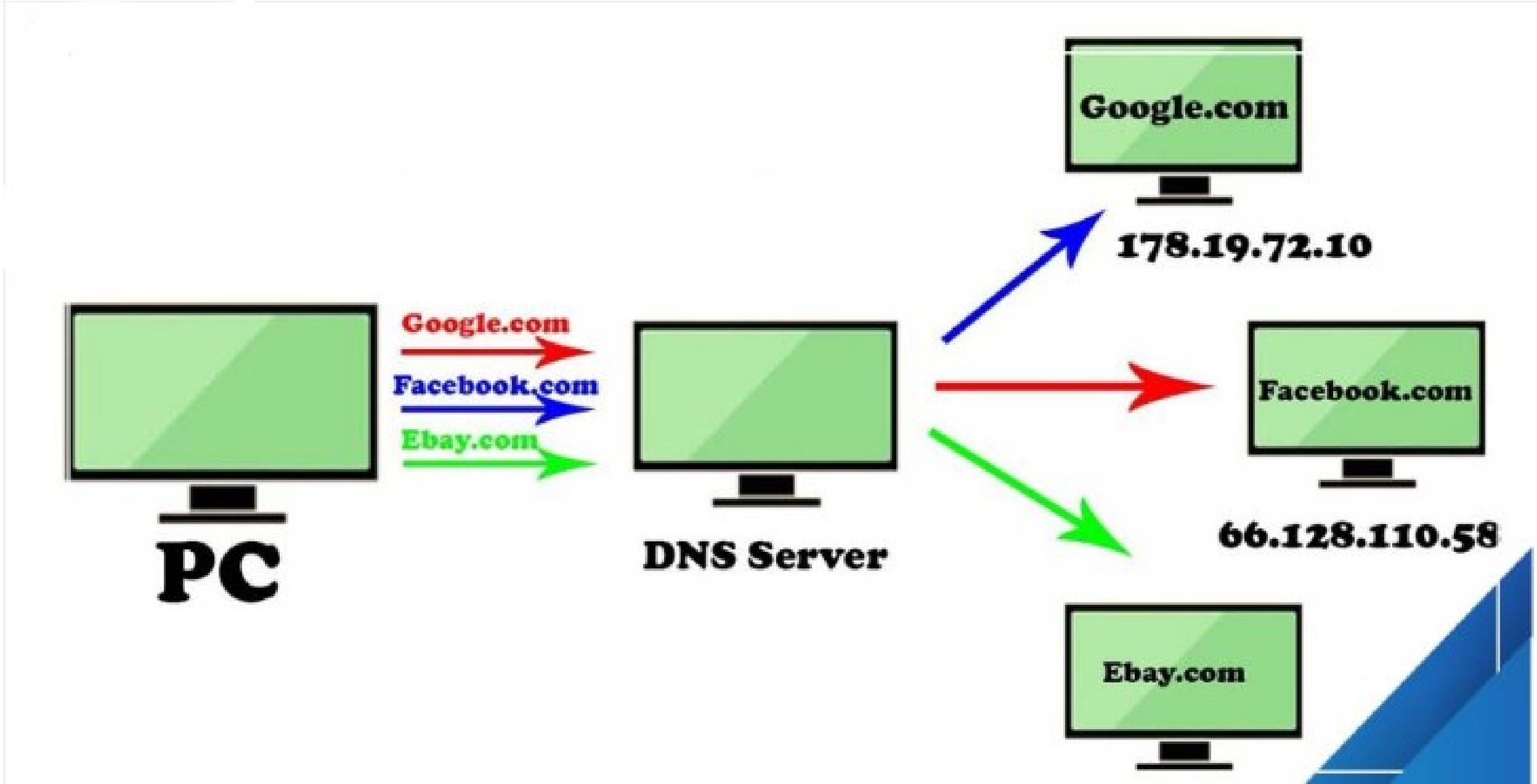
Ví dụ, www.example.com dịch thành 208.77.188.166.

tìm đến máy chủ nào đang quản lý tên miền, IP?



trả lời cho client IP tên miền mà máy chủ quản lý

Cơ chế vận hành của hệ thống DNS



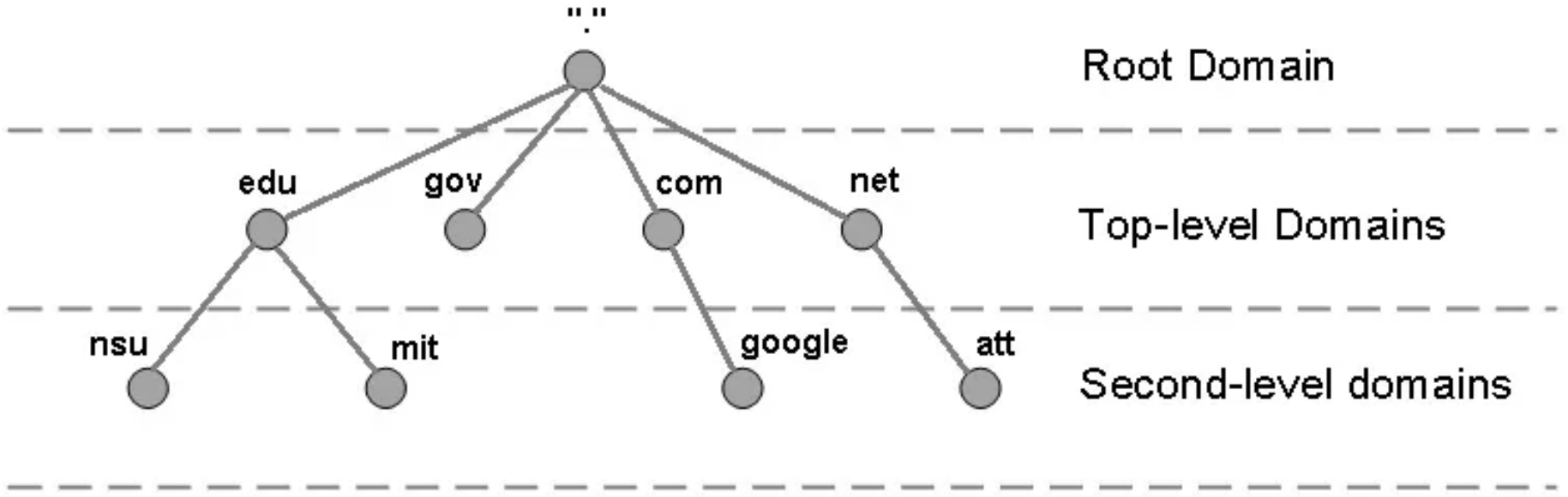
Quy tắc đặt tên miền

- Quy tắc đặt tên miền:
- Độ dài tối đa : 255 ký tự
- Độ dài tối đa của label : 63 ký tự
- Label phải bắt đầu bằng số hoặc chữ, chỉ chứa số, chữ, “-“, “.”
- Phân cấp tên miền : gốc, cấp 1, cấp 2…

DNS Namespace

– Hệ thống tên trong DNS sắp xếp theo mô hình phân cấp và cấu trúc cây logic được gọi là DNS namespace.

Cấu trúc của hệ thống tên miền



Biểu diễn đơn giản chỉ là dấu chấm “.”

– Tên miền cấp một (Top-level-domain) : gồm vài kí tự xác định một nước, khu vực hoặc tổ chức. “.com” , “.edu” ….

– Tên miền cấp hai (Second-level-domain): Nó rất đa dạng rất đa dạng có thể là tên một công ty, một tổ chức hay một cá nhân.

– Tên miền cấp nhỏ hơn (Subdomain) : Chia thêm ra của tên miền cấp hai trở xuống thường được sử dụng như chi nhánh, phòng ban của một cơ quan hay chủ đề nào đó.

Phân loại tên miền

– Com : Tên miền này được dùng cho các tổ chức thương mại.
– Edu : Tên miền này được dùng cho các cơ quan giáo dục, trường học.
– Net : Tên miền này được dùng cho các tổ chức mạng lớn.
– Gov : Tên miền này được dùng cho các tổ chức chính phủ.
– Org : Tên miền này được dùng cho các tổ chức khác.
– Int : Tên miền này dùng cho các tổ chức quốc tế.
– Info : Tên miền này dùng cho việc phục vụ thông tin.
– Arpa : Tên miền ngược.
– Mil : Tên miền dành cho các tổ chức quân sự, quốc phòng.
– Mã các nước trên thế giới tham gia vào mạng internet, các quốc gia này được qui định bằng hai chữ cái theo tiêu chuẩn ISO-3166 .Ví dụ : Việt Nam là .vn, Singapo là sg….

DNS Server

– DNS Server một máy tính có nhiệm vụ là DNS Server, chạy dịch vụ DNS service.

– DNS Server là một cơ sở dữ liệu chứa các thông tin về vị trí của các DNS domain và phân giải các truy vấn xuất phát từ các Client.

– DNS Server có thể cung cấp các thông tin do Client yêu cầu, và chuyển đến một DNS Server khác để nhờ phân giải hộ trong trường hợp nó không thể trả lời được các truy vấn về những tên miền không thuộc quyền quản lý và cũng luôn sẵn sàng trả lời các máy chủ khác về các tên miền mà nó quản lý.

- DNS Server lưu thông tin của Zone, truy vấn và trả kết quả cho DNS Client.

Hệ thống máy chủ DNS

- Máy chủ tên miền gốc (Root server)
- Trả lời truy vấn cho các máy chủ cục bộ
- Quản lý các zone và phân quyền quản lý cho máy chủ cấp dưới
- Có 13 hệ thống máy chủ gốc trên mạng Internet

(http://www.rootservers.org)

Hệ thống máy chủ DNS (tiếp)

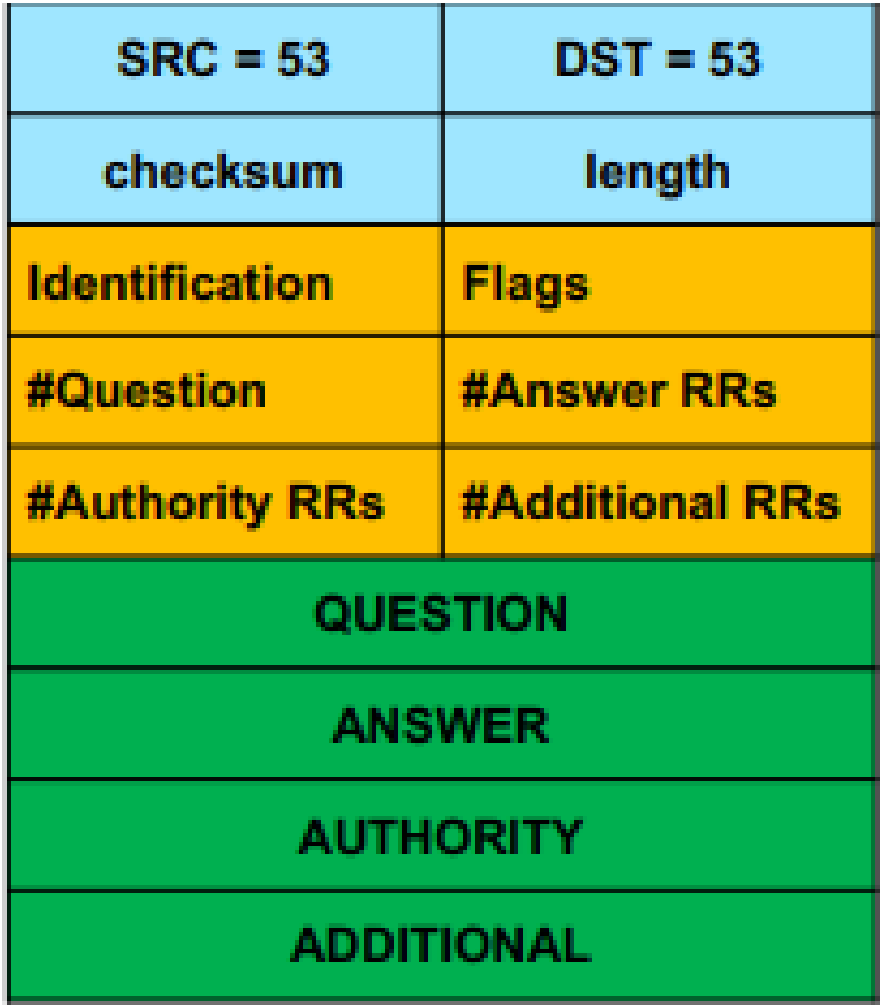
- Máy chủ tên miền cấp 1 (Top Level Domain)
- Quản lý tên miền cấp 1
- Máy chủ được ủy quyền (Authoritative DNS servers)
- Quản lý tên miền cấp dưới
- Máy chủ của các tổ chức: của ISP
- Máy chủ cục bộ: dành cho mạng nội bộ của cơ quan tổ chức
- Không nằm trong phân cấp của DNS

Phân giải tên miền

- Tự phân giải
- File HOST : C:\WINDOWS\system32\drivers\etc\
- Cache
- Dịch vụ phân giải tên miền DNS: client/server
- UDP, Port 53
- Phân giải đệ quy (Recursive Query)
- Phân giải tương tác (Interactive Query)

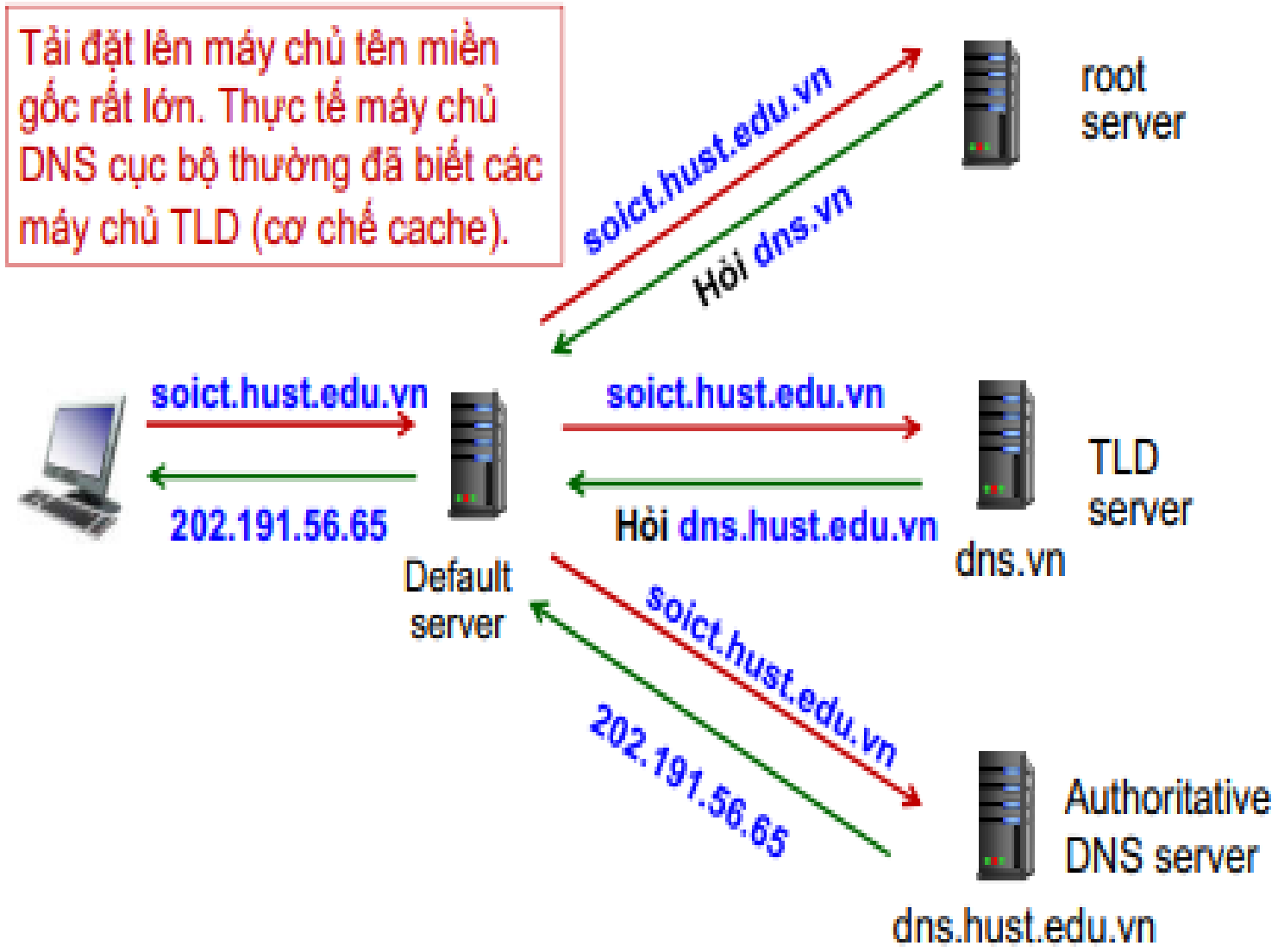
Thông điệp DNS

- DNS Query và DNS Reply
  - Chung khuôn dạng
- QUESTION: tên miền cần truy vấn
  - Số lượng: #Question
- ANSWER: thông tin tên miền tìm kiếm được
  - Số lượng: #Answer RRs
- AUTHORITY: địa chỉ server trả lời truy vấn
- ADDITIONAL: thông tin phân giải tên miền cho các địa chỉ khác



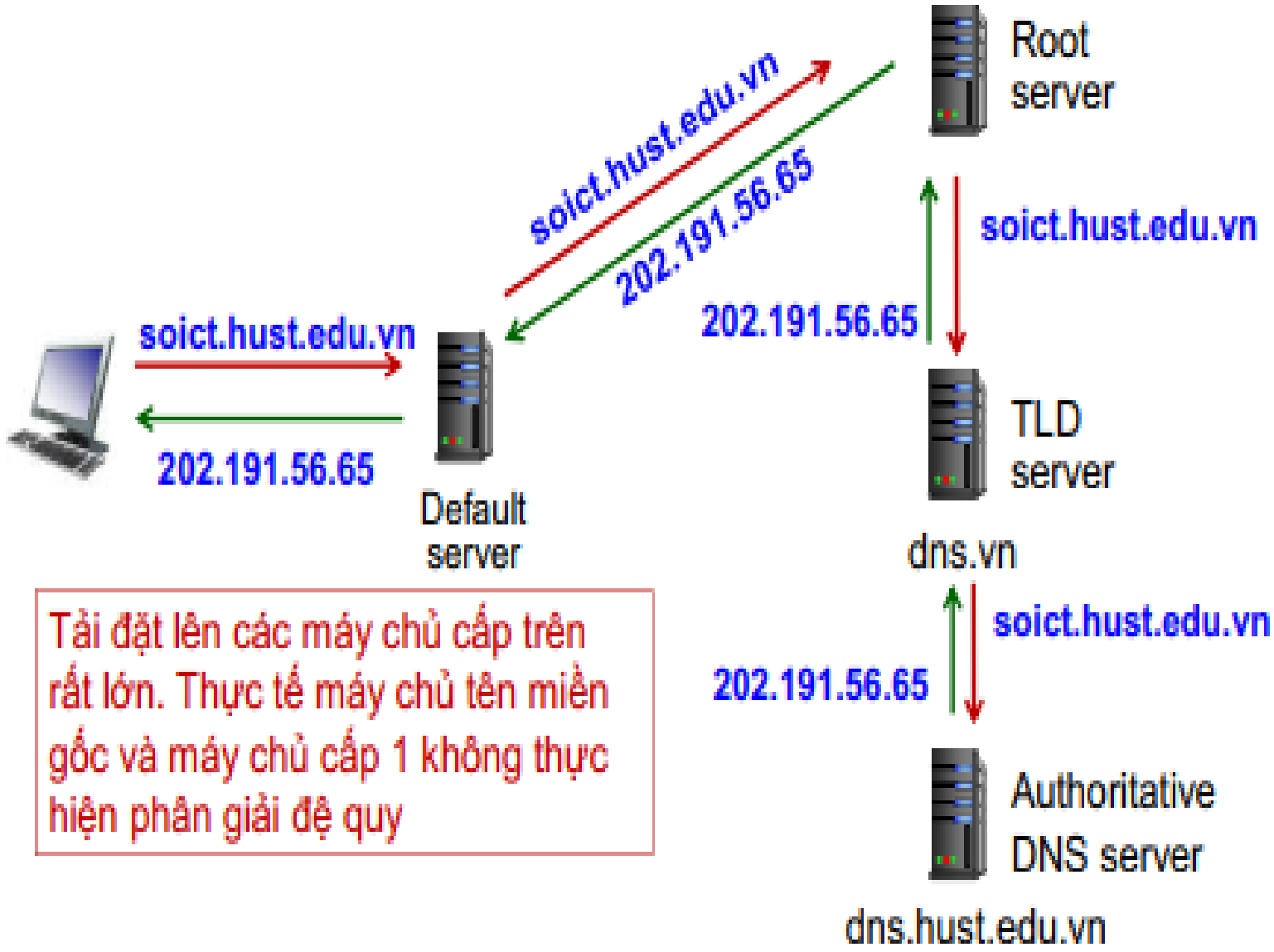
Phân giải tương tác

Cơ chế mặc định trên các máy chủ DNS

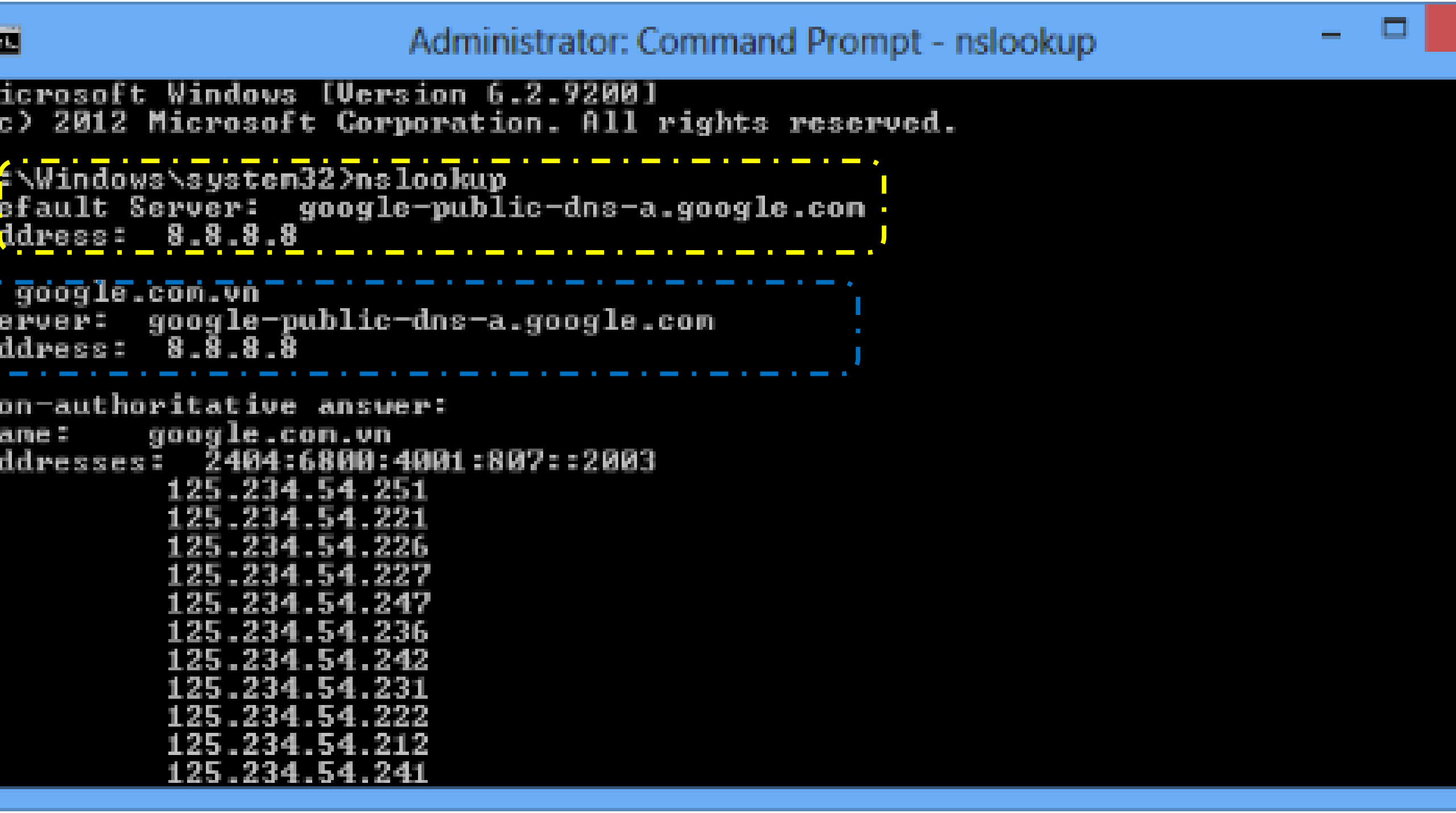


Phân giải đệ quy

Tùy chọn mở rộng



Tải đặt lên các máy chủ cấp trên rất lớn. Thực tế máy chủ tên miền gốc và máy chủ cấp 1 không thực hiện phân giải đệ quy


Administrator: Command Prompt - nslookup
icrosoft Windows [Version 6.2.9200]
c) 2012 Microsoft Corporation. All rights reserved.
:\Windows\system32>nslookup
efault Server: google-public-dns-a.google.com
ddress: 8.8.8.8
google.com.vn
erver: google-public-dns-a.google.com
ddress: 8.8.8.8
on-authoritative answer:
ame: google.com.vn
ddresses: 2404:6800:4001:807::2003
125.234.54.251
125.234.54.221
125.234.54.226
125.234.54.227
125.234.54.247
125.234.54.236
125.234.54.242
125.234.54.231
125.234.54.222
125.234.54.212
125.234.54.241

```
; <> DiG 9.9.2-P1 <> linux.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21655
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2,
ADDITIONAL: 3
;; QUESTION SECTION:
;linux.com.  IN A
;; ANSWER SECTION:
linux.com.  1786 IN A 140.211.167.51
linux.com.  1786 IN A 140.211.167.50
;; AUTHORITY SECTION:
linux.com.  86386 IN NS ns1.linux-foundation.org.
linux.com.  86386 IN NS ns2.linux-foundation.org.
;; ADDITIONAL SECTION:
ns1.linux-foundation.org.  261 IN A 140.211.169.10
ns2.linux-foundation.org.  262 IN A 140.211.169.11
```

Tên các máy chủ DNS server trả lời truy vấn. Nếu phần ANSWER rỗng, DNS Resolver gửi truy vấn tới các máy chủ này

```
; <> DiG 9.9.2-P1 <> linux.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21655
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2,
ADDITIONAL: 3
;; QUESTION SECTION:
;linux.com.  IN A
;; ANSWER SECTION:
linux.com. 1786 IN A 140.211.167.51
linux.com. 1786 IN A 140.211.167.50
;; AUTHORITY SECTION:
linux.com. 86386 IN NS ns1.linux-foundation.org.
linux.com. 86386 IN NS ns2.linux-foundation.org.
;; ADDITIONAL SECTION:
ns1.linux-foundation.org. 261 IN A 140.211.169.10
ns2.linux-foundation.org. 262 IN A 140.211.169.11
```

Địa chỉ IP của các máy chủ trả lời truy vấn.
Thông tin này được lưu vào cache

3. HTTP và WWW

HTTP và Web

- Internet trước thập kỷ 1990s:
- Hầu như chỉ sử dụng hạn chế trong cơ quan chính phủ, phòng nghiên cứu...
- Các dịch vụ email, FPT không phù hợp cho chia sẻ thông tin đại chúng
- Không có cơ chế hiệu quả để liên kết các tài nguyên thông tin nằm rải rác trên Internet
- Năm 1990, Tim Berners-Lee giới thiệu World Wide Web:
- Trao đổi thông tin dưới dạng siêu văn bản (hypertext) sử dụng ngôn ngữ.

- Http (HyperText Transfer Protocol) là giao thức truyền tải siêu văn bản được sử dụng trong www dùng để truyền tải dữ liệu giữa Web server đến các trình duyệt Web và ngược lại. Giao thức này sử dụng cổng 80 (port 80) là chủ yếu.

Uniform Resource Locator

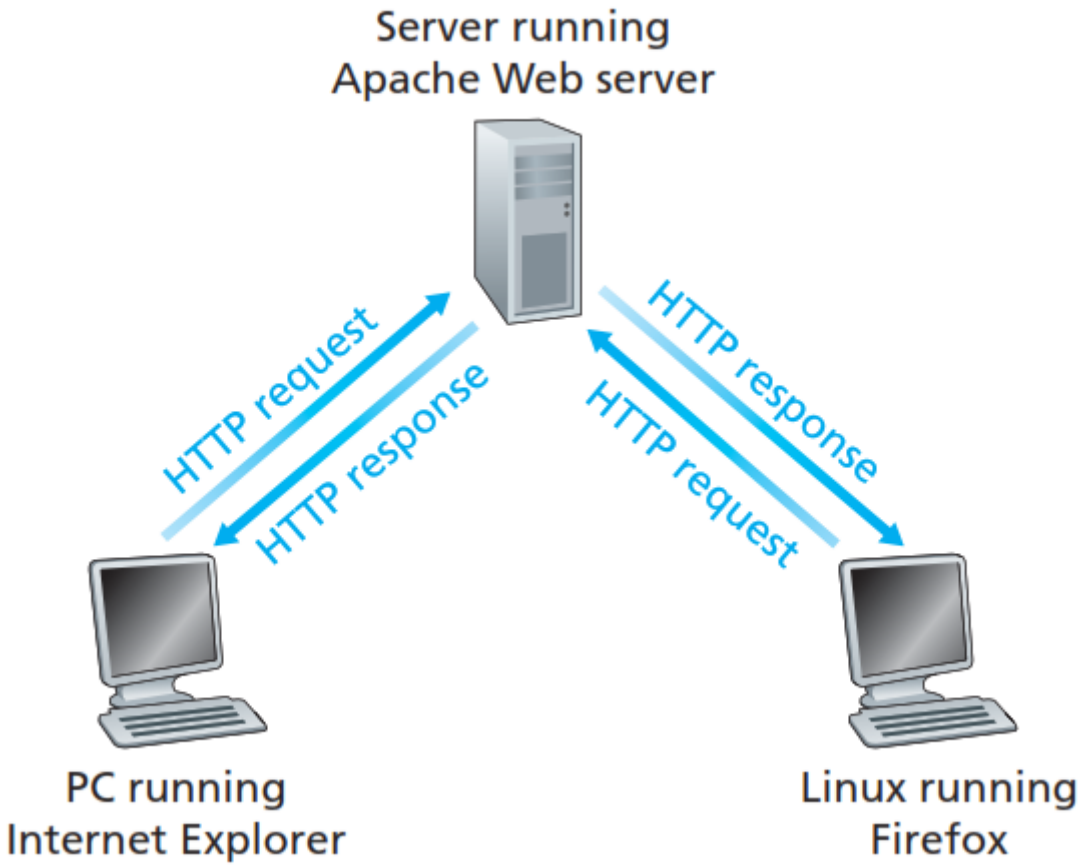
- Định vị một tài nguyên bất kỳ trên mạng và cách thức để truy cập tài nguyên đó

`protocol://hostname[:port]/directory-path/resource`

- *protocol*: Giao thức (http, ftp, https, smtp, rtsp…)
- *hostname*: tên miền, địa chỉ IP
- *port:* cổng ứng dụng (có thể không cần)
- *directory path*: đường dẫn tới tài nguyên
- *resource*: định danh của tài nguyên

HTTP và Web

- WWW: World Wide Web
  - trao đổi dữ liệu siêu văn bản HTML (HyperText Markup Language) trên mạng
- HTTP: HyperText Transfer Protocol
  - Mô hình Client/Server
  - Client yêu cầu truy nhập tới các trang web (chứa các đối tượng web) và hiển thị chúng trên trình duyệt
  - Server: Nhận yêu cầu và trả lời cho client



HTTP hoạt động ntn?

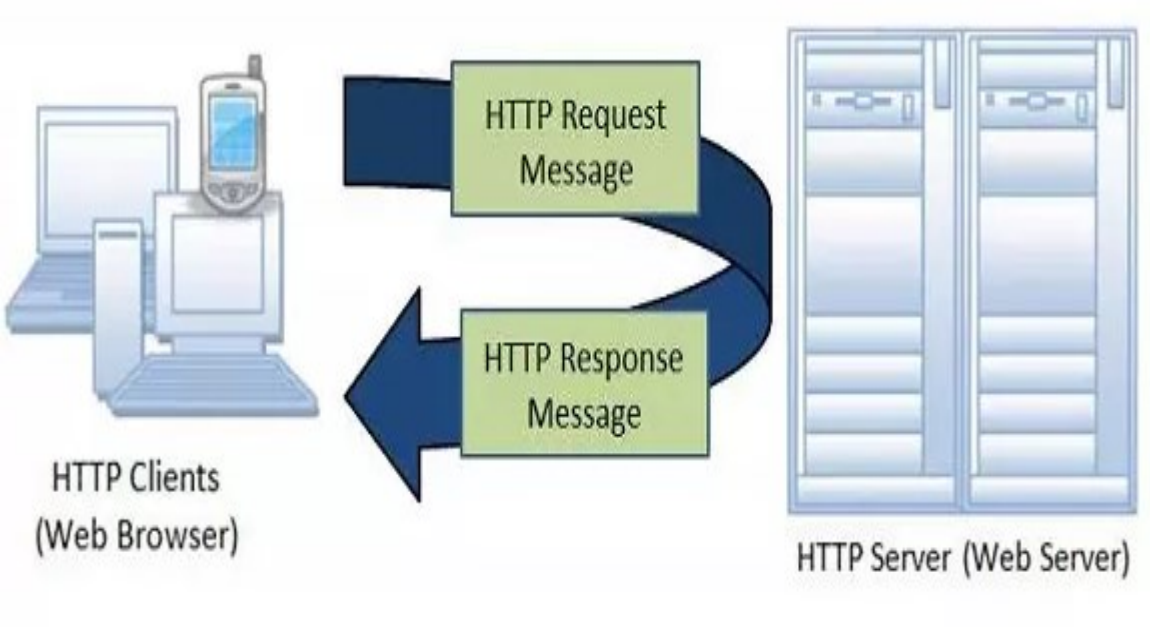
- Server mở một TCP socket chờ yêu cầu kết nối tại cổng 80 (default)

- Client khởi tạo một liên kết TCP tới server

- Server chấp nhận yêu cầu, tạo liên kết

- Trao đổi thông điệp HTTP (giao thức ứng dụng)
  - HTTP Request
  - HTTP Response

- Đóng liên kết TCP

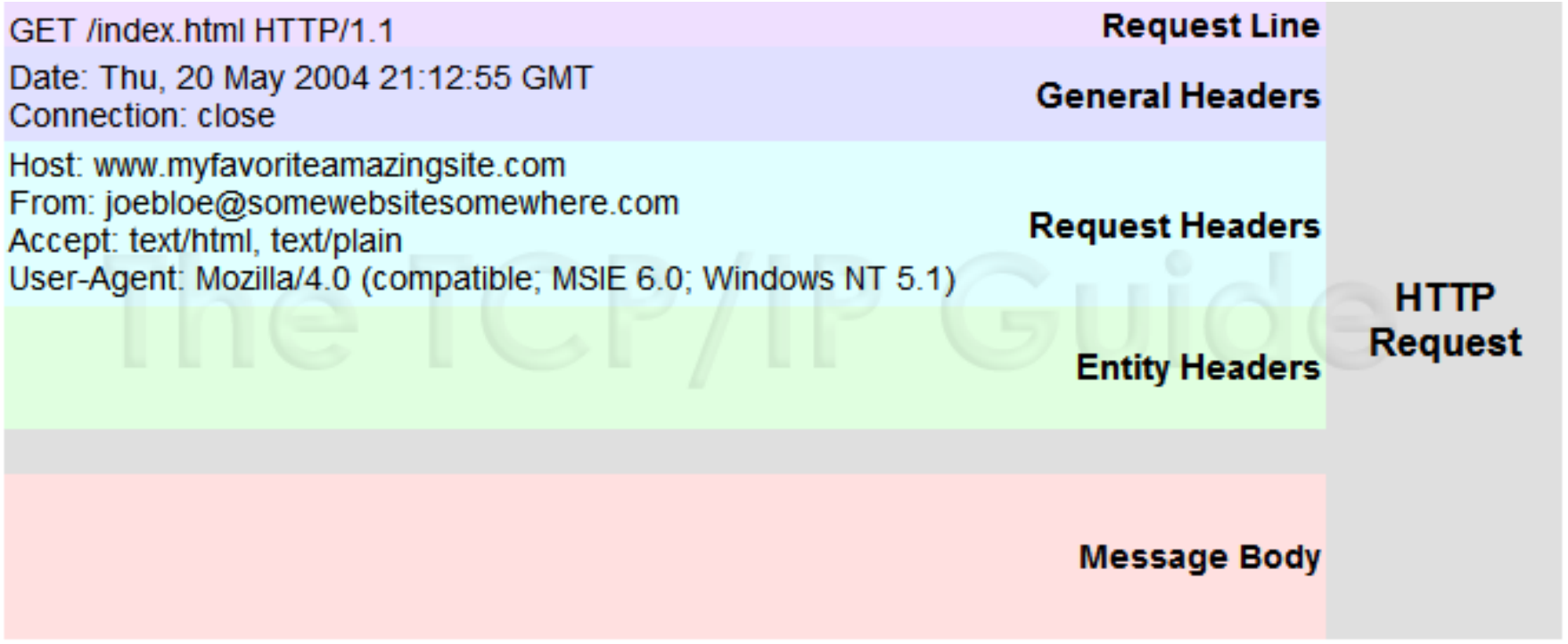


Format gói tin:

Trong suốt phiên làm việc HTTP , có hai loại gói tin chính được trao đổi là gói tin yêu cầu gửi từ client đến server gọi là "Request message" và gói tin phản hồi gửi từ server đến client gọi là "Response message". Hai loại gói tin này được giao thức HTTP định nghĩa rất cụ thể:

HTTP request message

Gồm 3 phần chính là: Request line, Header và Body. Dữ liệu được chia thành các dòng và định dạng kết thúc dòng là <CR><LF> (tương ứng với các ký tự 0x0A,0x0D trong bảng mã ASCII).



•Request line là dòng đầu tiên của gói HTTP Request, bao gồm 3 trường: phương thức (method), đường dẫn (path – URL hoặc URI cho trường này) và (HTTP version).

- **Phương thức (method)** có thể là: GET, POST, HEAD, PUT và DELETE. Hai phương thức phổ biến nhất là GET và POST, GET thường dùng để yêu cầu tài nguyên cung cấp trong trường URL.
- **Đường dẫn (path)** dùng để định danh nguồn tài nguyên mà client yêu cầu, bắt buộc phải có ít nhất là dấu "/".
- **Phiên bản giao thức (HTTP version):** là phiên bản HTTP client đang sử dụng (thường là HTTP/1.0 hoặc HTTP/1.1)

• Header, các dòng này là không bắt buộc, viết ở định dạng "Name:Value" cho phép client gửi thêm các thông tin bổ sung về thông điệp HTTP request và thông tin về chính client. Một số header thông dụng như:

- **Accept**: nội dung có thể nhận được từ thông điệp response. Ví dụ: text/plain, text/html
- **Accept-Encoding**: các kiểu nén được chấp nhận. Ví dụ: gzip, deflate, xz, exi…
- **Connection**: tùy chọn điều khiển cho kết nối hiện thời. Ví dụ: Keep-Alive, Close…
- **Cookie**: thông tin HTTP Cookie từ server

•Body, là dữ liệu gửi từ client đến server trong gói tin HTTP request. Đa số các gói tin gửi theo phương thức GET sẽ có Body trống, các phương thức như POST hay PUT thường dùng để gửi dữ liệu nên sẽ có bao gồm dữ liệu trong trường Body

HTTP response message

Định dạng gói tin HTTP response cũng gồm 3 phần chính là: Status line, Header và Body.

HTTP/1.1 200 OK	**Status Line**	**HTTP Response**
Date: Thu, 20 May 2004 21:12:58 GMT Connection: close	**General Headers**	
Server: Apache/1.3.27 Accept-Ranges: bytes	**Response Headers**	
Content-Type: text/html Content-Length: 170 Last-Modified: Tue, 18 May 2004 10:14:49 GMT	**Entity Headers**	
<html> <head> <title>Welcome to the Amazing Site!</title> </head> <body> This site is under construction. Please come back later. Sorry! </body> </html>	**Message Body**	

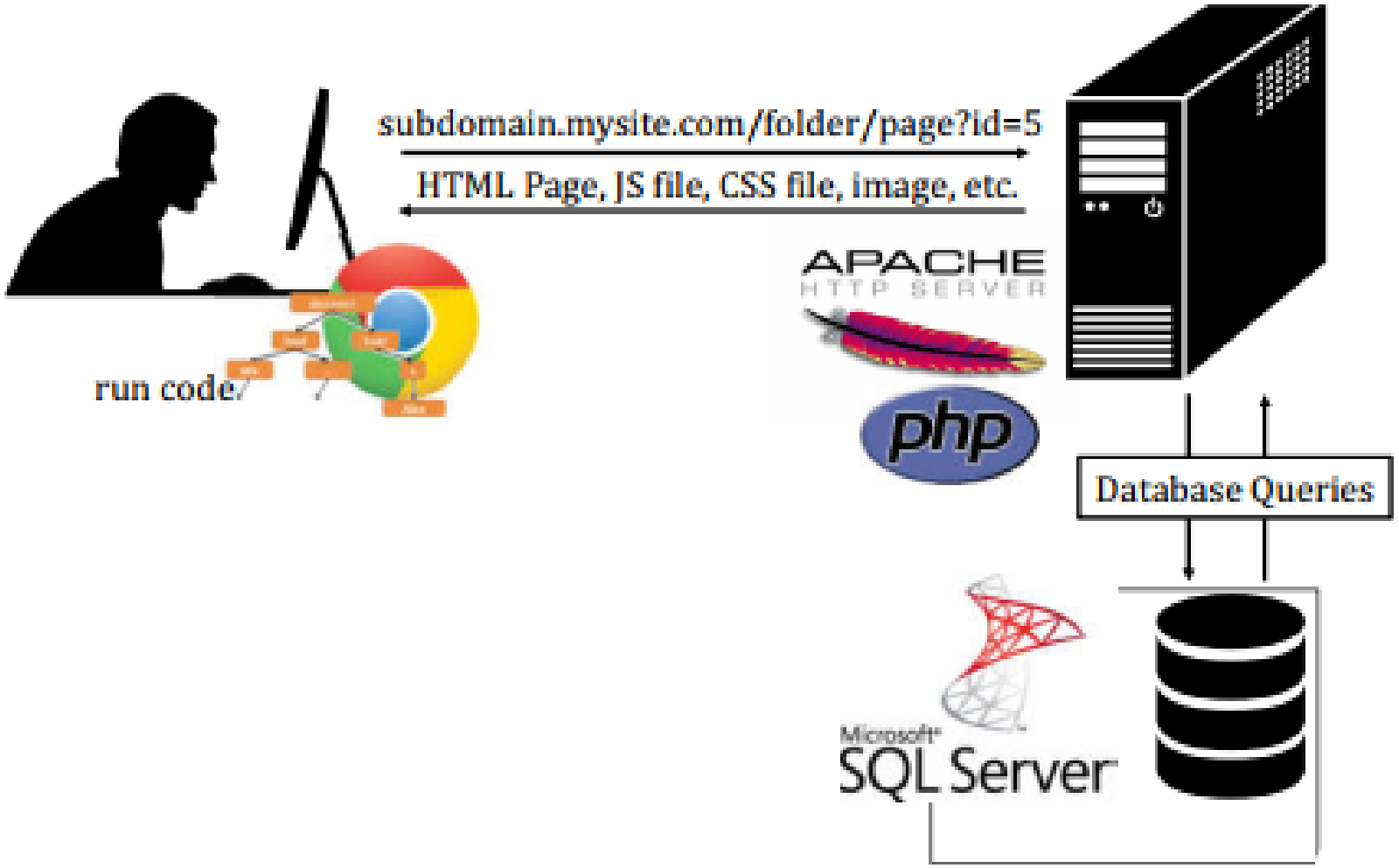
•Dòng Status line gồm 3 trường là phiên bản giao thức (HTTP version), mã trạng thái (Status code) và mô tả trạng thái (Status text):

- **Phiên bản giao thức (HTTP version)**: phiên bản của giao thức HTTP mà server hỗ trợ, thường là HTTP/1.0 hoặc HTTP/1.1
- **Mã trạng thái (Status code)**: mô tả trạng thái kết nối dưới dạng số, mỗi trạng thái sẽ được biểu thị bởi một số nguyên. Ví dụ: 200, 404, 302,…
- **Mô tả trạng thái (Status text)**: mô tả trạng thái kết nối dưới dạng văn bản một cách ngắn gọn, giúp người dùng dễ hiểu hơn so với mã trạng thái. Ví du: 200 OK, 404 Not Found, 403 Forbiden,…

•Header line của gói tin response có chức năng tương tự như gói tin request, giúp server có thể truyền thêm các thông tin bổ sung đến client dưới dạng các cặp *"Name:Value"*.

•Phần Body là nơi đóng gói dữ liệu để trả về cho client, thông thường trong duyệt web thì dữ liệu trả về sẽ ở dưới dạng một trang HTML để trình duyệt có thể thông dịch được và hiển thị ra cho người dùng.

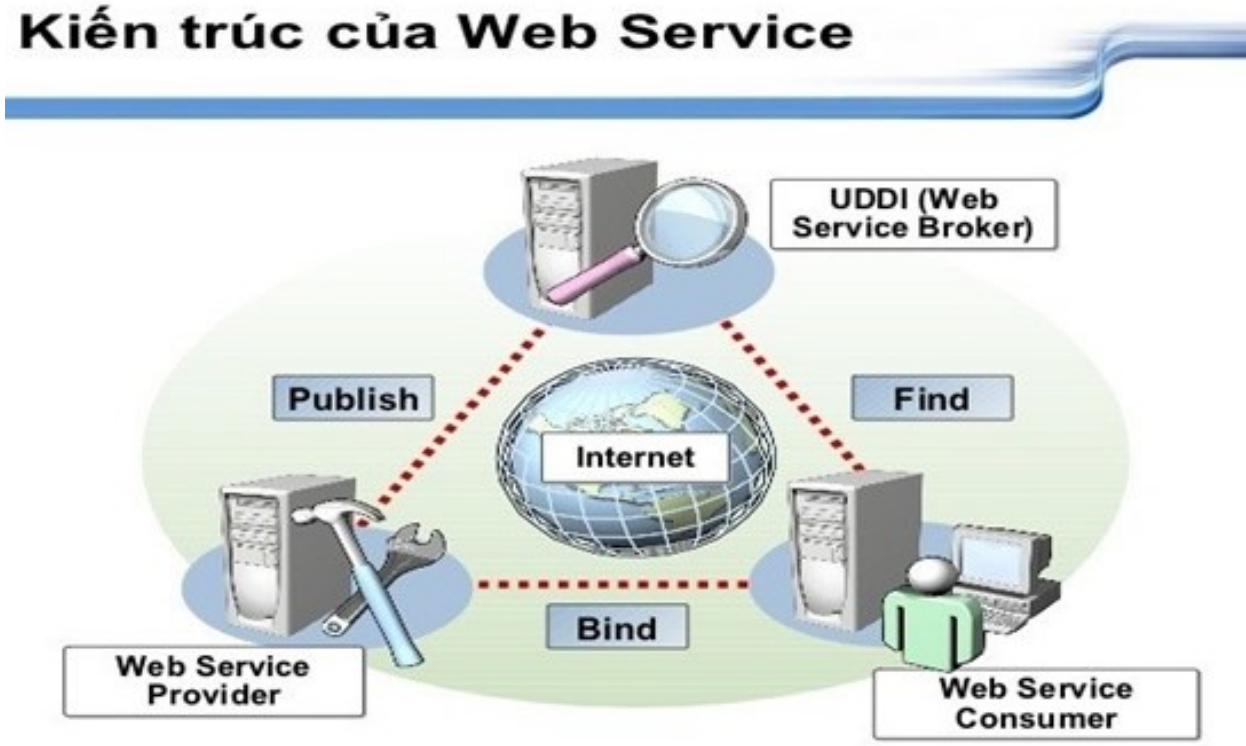
Kiến trúc chung của các dịch vụ web



Protocol Stack tại Web Service

Stack vẫn đang được phát triển và hiện tại có bốn lớp chính là:

- Service Transport
- XML Messaging
- Service Description
- Service Discovery.



•**Service Transport:** ở lớp này sẽ chịu trách nhiệm vận chuyển thông tin giữa các ứng dụng. Hiện tại, lớp này sẽ có các thành phần như: giao thức truyền tải văn bản Hyper (HTTP), giao thức thư đơn giản (SMTP), giao thức truyền tệp (FTP), giao thức trao đổi mở rộng khối (BEEP).

•**XML Messaging:** ở lớp này sẽ có trách nhiệm mã hóa những thông điệp theo định dạng XML phổ biến, đảm bảo có thể hiểu các thông điệp ở hai đầu. Lớp này bao gồm các yếu tố: XML – RPC, SOAP.

•**Service Description:** ở lớp này sẽ có trách nhiệm mô tả giao diện công cộng cho một Web Service cụ thể. Hiện tại, các mô tả dịch vụ được xử lý thông qua ngôn ngữ mô tả Web Service là WSDL.

•**Service Discovery:** ở lớp này sẽ có nhiệm vụ tập trung những dịch vụ vào một số đăng ký chung và cung cấp chức năng xuất bản/ tìm kiếm dễ dàng. Hiện tại, việc khám phá dịch vụ sẽ được xử lý thông qua việc mô tả chung, khám phá và tích hợp (UDDI).

Hiển thị (rendering) nội dung trang web

- Mô hình xử lý cơ bản tại trình duyệt: mỗi cửa sổ hoặc 1 frame:
- Nhận thông điệp HTTP Response
- Hiển thị:
- Xử lý mã HTML, CSS, Javascripts
- Gửi thông điệp HTTP Request yêu cầu các đối tượng khác(nếu có)
- Bắt và xử lý sự kiện
- Các sự kiện có thể xảy ra:
- Sự kiện của người dùng: OnClick, OnMouseOver…
- Sự kiện khi hiển thị: OnLoad, OnBeforeUnload…
- Theo thời gian: setTimeout(), clearTimeout()…

Giao thức HTTP

- Hypertext Transfer Protocol (HTTP) là giao thức thuộc lớp ứng dụng web. Sử dụng kết nối TCP cổng 80. HTTP được định nghĩa trong RFC 1945 (HTTP 1.0) và RFC 2616 (HTTP 1.1).

- HTTP hoạt động dựa trên mô hình client-server. Trình duyệt client thực hiện yêu cầu, nhận và hiển thị đối tượng web (gồm dữ liệu HTML, hình ảnh JPEG, Java applet, video, âm thanh, ...). Trong khi, web server sẽ gửi trả lời khi nhận được yêu cầu từ client.

Kết nối HTTP

Có hai loại kết nối HTTP là kết nối không bền vững và kết nối bền vững.

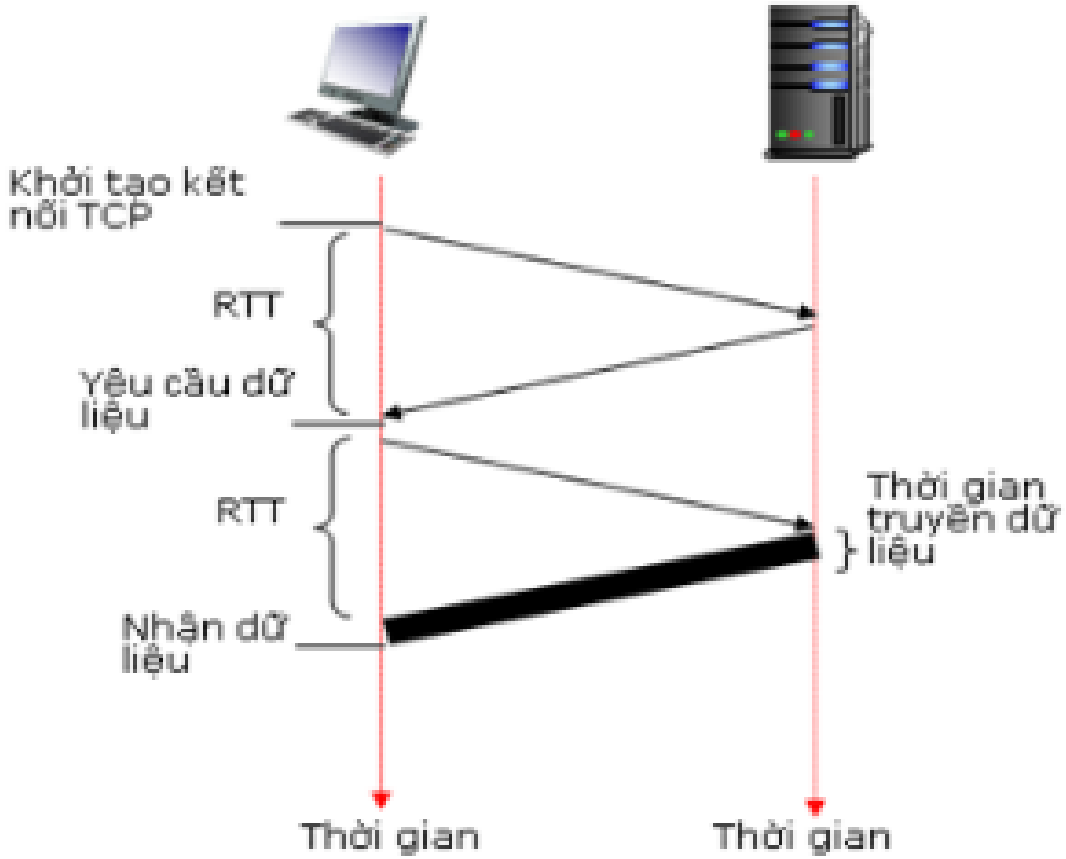
- **Kết nối không bền vững:** sau khi, server gửi đi một đối tượng thì kết nối TCP sẽ được đóng. Như vậy, mỗi kết nối TCP chỉ truyền được duy nhất một yêu cầu từ client và nhận lại một thông điệp trả lời từ server.

- **Kết nối bền vững**: server sẽ duy trì kết nối TCP cho việc gửi nhiều đối tượng. Như vậy, sẽ có nhiều yêu cầu từ client được gửi đến server trên cùng một kết nối. Thông thường kết nối TCP này sẽ được đóng lại trong một khoảng thời gian định trước.

Quy trình hoạt động của kết nối HTTP không bền vững:

Bước 1: client khởi tạo kết nối TCP bằng việc gửi yêu cầu đến server. Server nhận được yêu cầu và chấp nhận kết nối bằng việc gửi trả lời về cho client. Nếu sau khoảng thời gian RTT mà không nhận được trả lời từ phía server thì client sẽ gửi lại yêu cầu.

- Bước 2: sau khi kết nối được thiết lập, client sẽ gửi thông điệp yêu cầu chứa tên đường dẫn của các đối tượng (ví dụ: www.hutech.edu.vn/homepage/index.php) đến server. Server nhận được thông điệp yêu cầu và tiến hành lấy ra các đối tượng được yêu cầu. Sau đó, các đối tượng được đóng gói thành thông điệp trả lời và gửi đến client.



- Bước 3: server đóng kết nối TCP (Lưu ý: server chỉ đóng kết nối TCP khi chắc chắn rằng client nhận được thông điệp trả lời)
- Bước 4: client nhận thông điệp trả lời chứa tập tin HTML và hiển thị các đối tượng.
- Bước 5: lặp lại các bước từ 1-4 cho các đối tượng khác.

Lưu ý: Trong kết nối HTTP không bền vững cần có một RTT (Round Trip Time) để khởi tạo kết nối TCP. Ngoài ra, cần có một RTT cho thông điệp HTTP yêu cầu và một vài byte đầu tiên của thông điệp HTTP trả lời được trả về.

Tổng thời gian truyền tập tin = 2RTT+ thời gian truyền.

Thời gian đáp ứng RTT là thời gian gửi một gói tin cơ bản từ client đến server rồi quay ngược lại. RTT bao gồm độ trễ truyền gói tin và hàng đợi, độ trễ trong các bộ định tuyến trung gian, chuyển mạch và xử lý gói tin.

Qui trình hoạt động của kết nối HTTP bền vững:

- Kết nối bền vững không có pipelining:

Client phát ra yêu cầu mới, chỉ khi đáp ứng trước đó đã nhận xong.

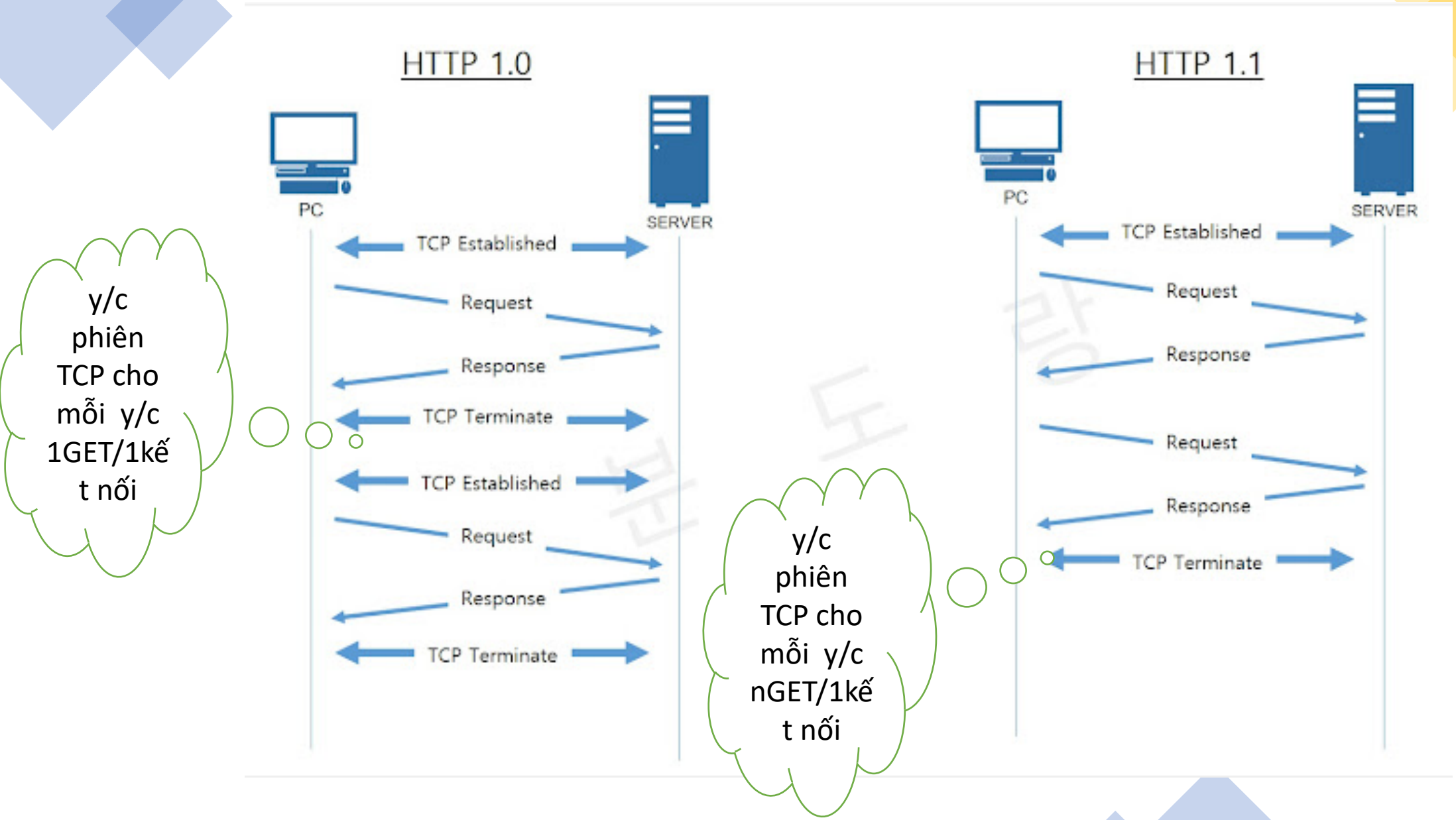
RTT cho mỗi đối tượng tham chiếu.

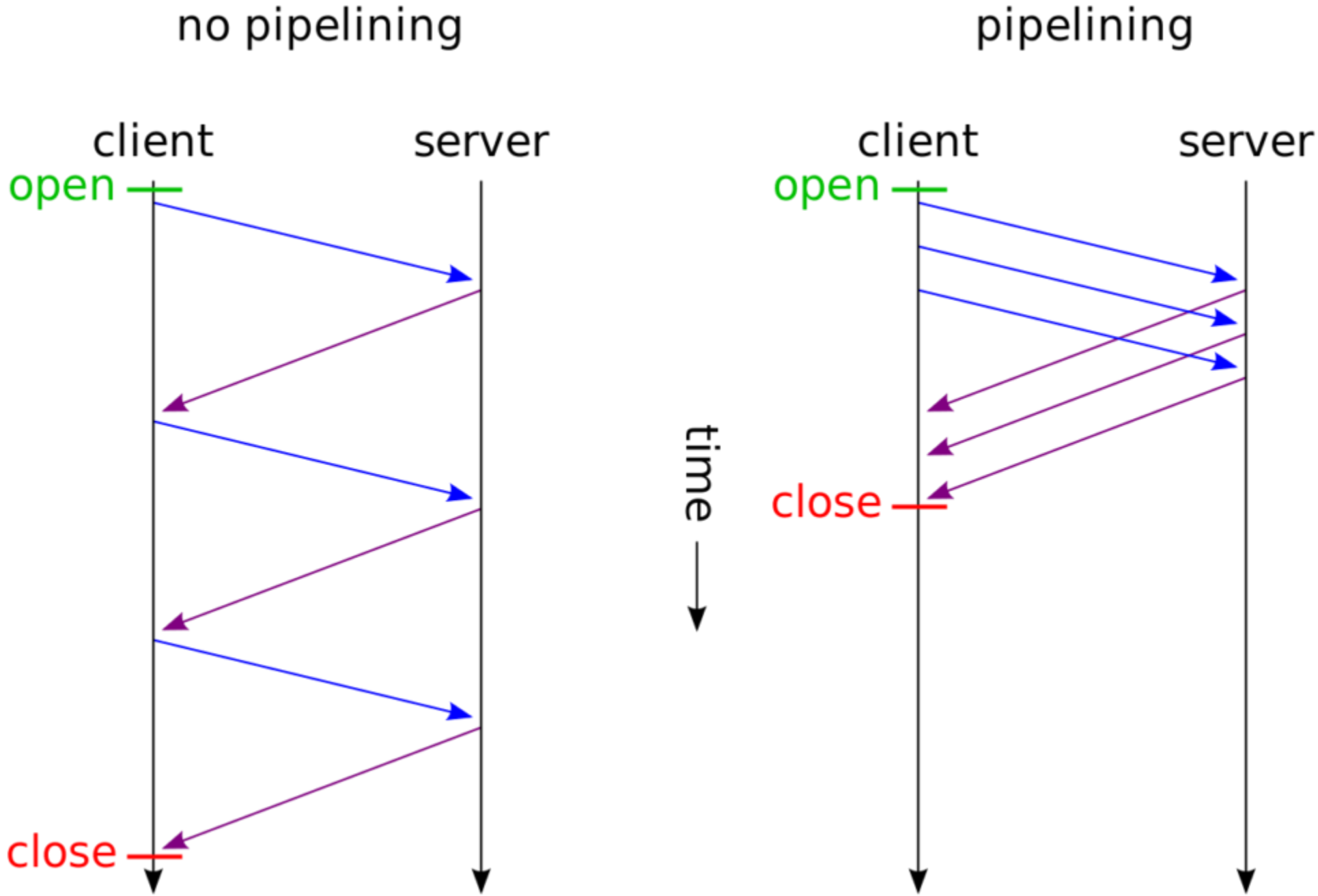
- Kết nối bền vững có pipelining:

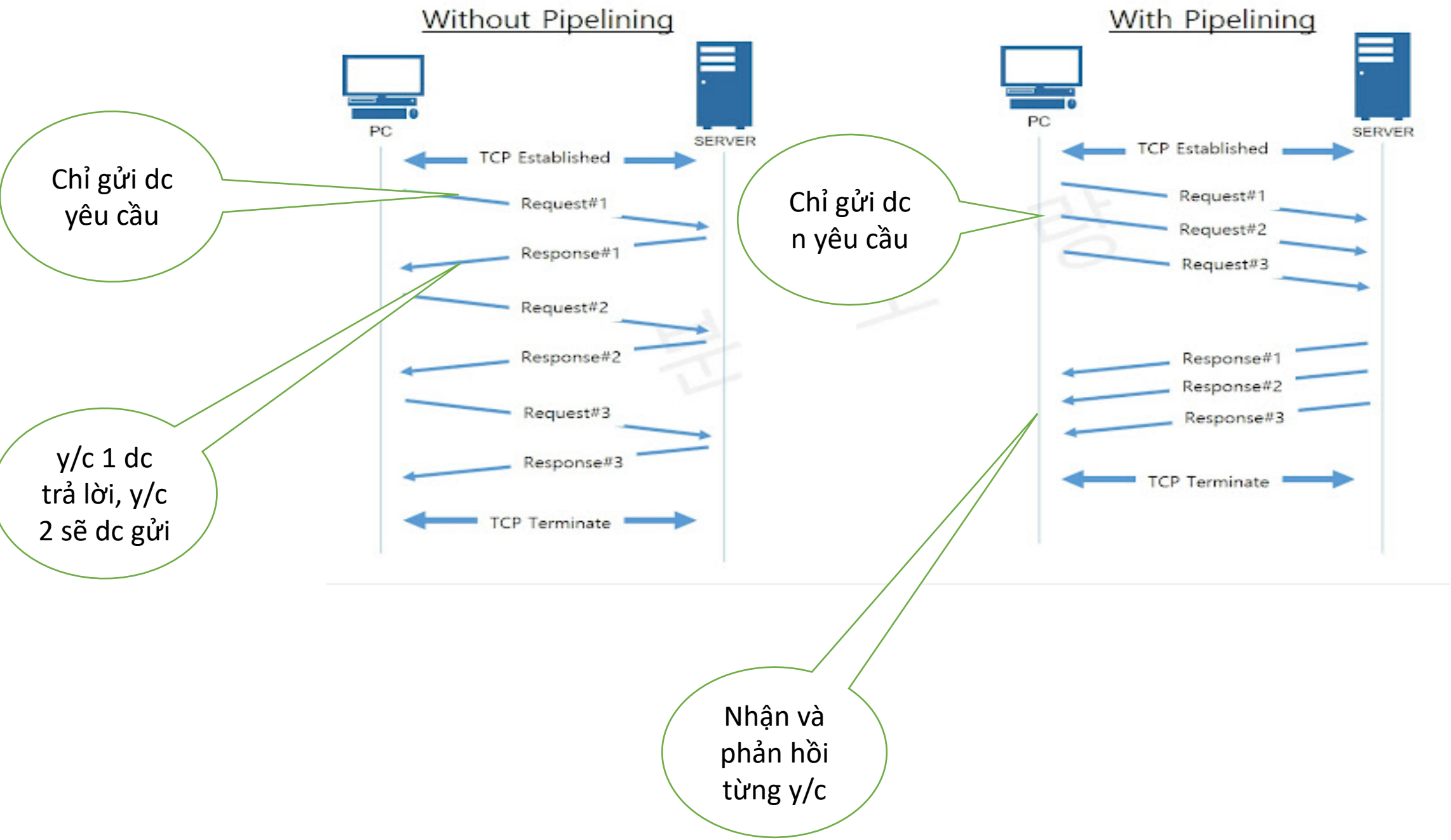
Mặc định có trong HTTP/1.1.

Client gửi yêu cầu ngay sau khi gặp một đối tượng tham chiếu.

Ít nhất 1 RTT cho tất cả đối tượng tham chiếu.


HTTP 1.0
PC
SERVER
TCP Established
Request
Response
TCP Terminate
TCP Established
Request
Response
TCP Terminate
y/c phiên TCP cho mỗi y/c 1GET/1kết nối
HTTP 1.1
PC
SERVER
TCP Established
Request
Response
Request
Response
TCP Terminate
y/c phiên TCP cho mỗi y/c nGET/1kết nối


no pipelining
client
server
open
close
time
pipelining
client
server
open
close


Without Pipelining
PC
SERVER
TCP Established
Request#1
Response#1
Request#2
Response#2
Request#3
Response#3
TCP Terminate
With Pipelining
PC
SERVER
TCP Established
Request#1
Request#2
Request#3
Response#1
Response#2
Response#3
TCP Terminate
Chỉ gửi dc yêu cầu
y/c 1 dc trả lời, y/c 2 sẽ dc gửi
Chỉ gửi dc n yêu cầu
Nhận và phản hồi từng y/c

HTTP Cookie

HTTP Cookie được định nghĩa như sau:

HTTP Cookie (web cookie, browser cookie) là dữ liệu được gửi từ server tới trình duyệt của người dùng. Trình duyệt sẽ lưu dữ liệu cookie này và gửi lại theo mỗi HTTP request về cho cùng server đó. Về cơ bản, cookie dùng để nói cho server biết các request đến từ một trình duyệt, ví dụ để giữ lại trạng thái đăng nhập...

HTTP là stateless. Do đó, mọi request đến server đều giống nhau. Vì vậy, server không thể phân biệt được request được gửi đến là từ một client đã thực hiện request trước đó, đã đăng nhập,... hay từ một client mới.

Vì vậy, HTTP Cookie đã ra đời để giải quyết vấn đề này.

Các tác dụng của cookie

Cụ thể, các tác dụng chính của cookie là:

- **Quản lý session**: trạng thái đăng nhập, thông tin giỏ hàng,...

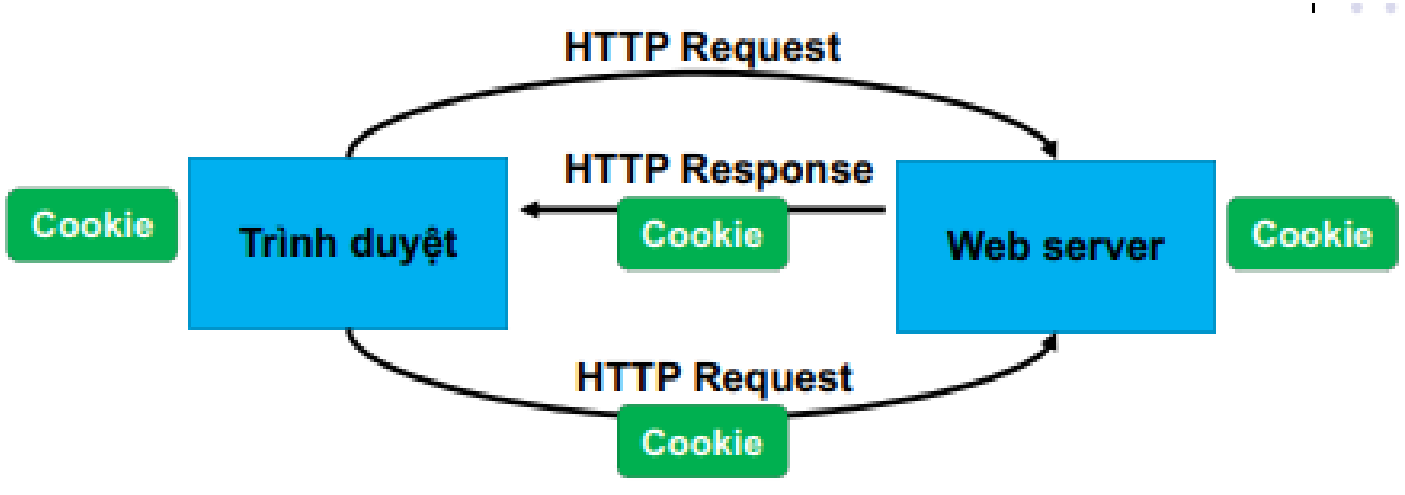
- **Lưu thông tin cài đặt của người dùng**: theme (dark mode hay light mode), ngôn ngữ (tiếng việt, tiếng anh), thông tin username/email trong mục bình luận,...

- **Theo dõi và phân tích hành vi người dùng**: các trang đã truy cập, truy cập bao nhiêu lần,...

HTTP Cookie có nhiều lợi ích, nhưng cũng tồn tại nhiều giới hạn như:

- Dung lượng lưu trữ giới hạn khoảng 4KB. Vì vậy, số lượng cookie lưu trữ cũng giới hạn theo.
- Cookie là **private** đối với một domain. Do đó, một trang web chỉ có thể đọc được cookies của chính nó.
- Khi số lượng cookie được lưu trữ vượt quá giới hạn, các cookie mới tạo ra sẽ thay thế cookies cũ.
- Nếu người dùng **disable cookie** thì bạn sẽ không thể sử dụng được các tính năng của nó.
- Thông tin lưu trữ ở cookie là không bảo mật nên có thể bị đánh cắp.

HTTP Cookie



• Cookie: dữ liệu do ứng dụng Web tạo ra, chứa thông tin trạng thái của phiên làm việc Server có thể lưu lại cookie(một phần hoặc toàn bộ)
- Sau khi xử lý yêu cầu, Web server trả lại thông điệp HTTP Response với cookie đính kèm
- Set-Cookie: key = value; options;
- Trình duyệt lưu cookie
- Trình duyệt gửi HTTP Request tiếp theo với cookie được đính kèm

HTTP Cookie - Ví dụ

HTTP Response

```
HTTP/1.1 200 OK
Server: nginx/1.10.1
Date: Mon, 14 Nov 2016 09:19:19 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.4.45
Set-Cookie: vflastvisit=1479115159; expires=Tue, 14-Nov-2017 09:19:19 GMT; path=/; domain=vozforums.com; secure
Set-Cookie: vflastactivity=0; expires=Tue, 14-Nov-2017 09:19:19 GMT; path=/; domain=vozforums.com; secure
Expires: 0
Cache-Control: private, post-check=0, pre-check=0, max-age=0
Pragma: no-cache
Content-Encoding: gzip
```

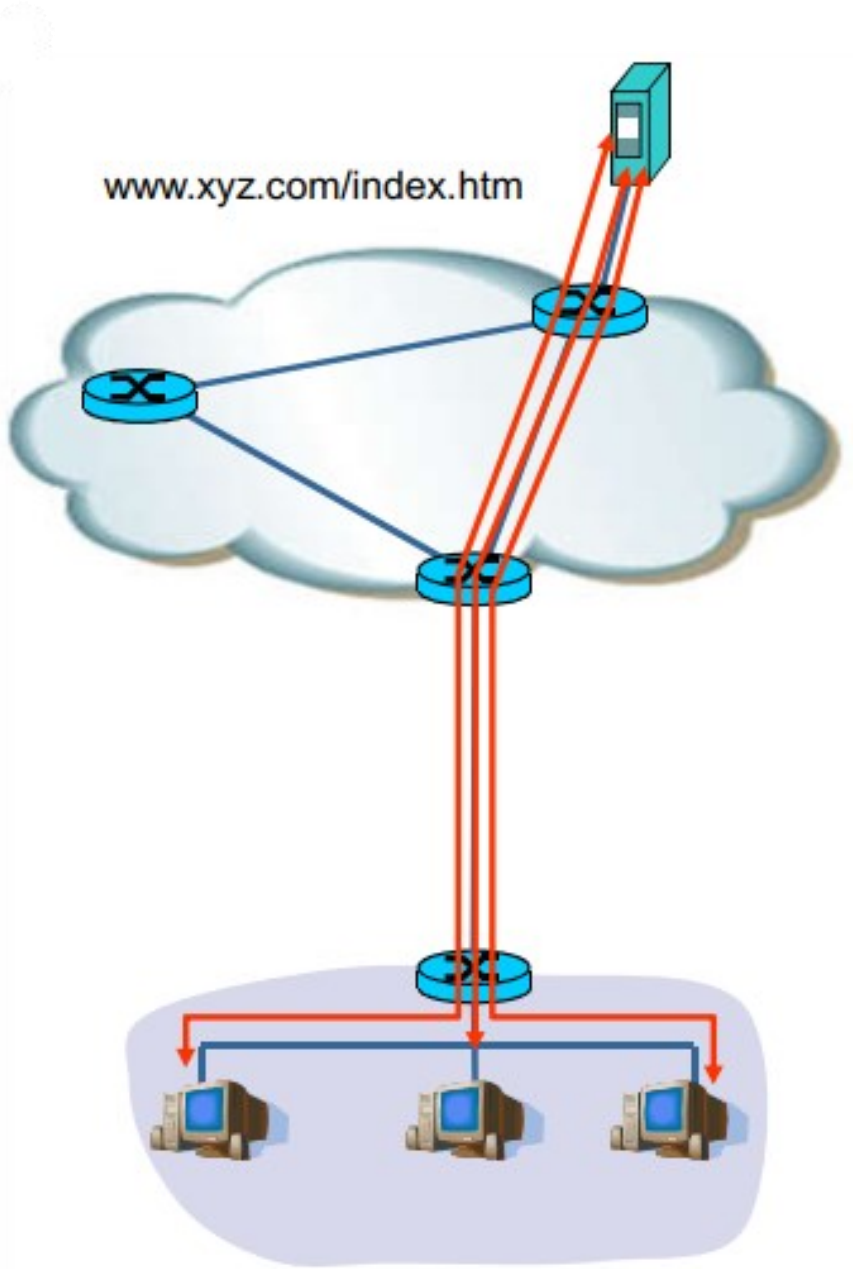
HTTP Cookie - Ví dụ

- HTTP Request

```
GET /clientscript/vbulletin_important.css?v=380 HTTP/1.1
Host: vozforums.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://vozforums.com/
Cookie: vflastvisit=1479115159; vflastactivity=0
Connection: keep-alive
```

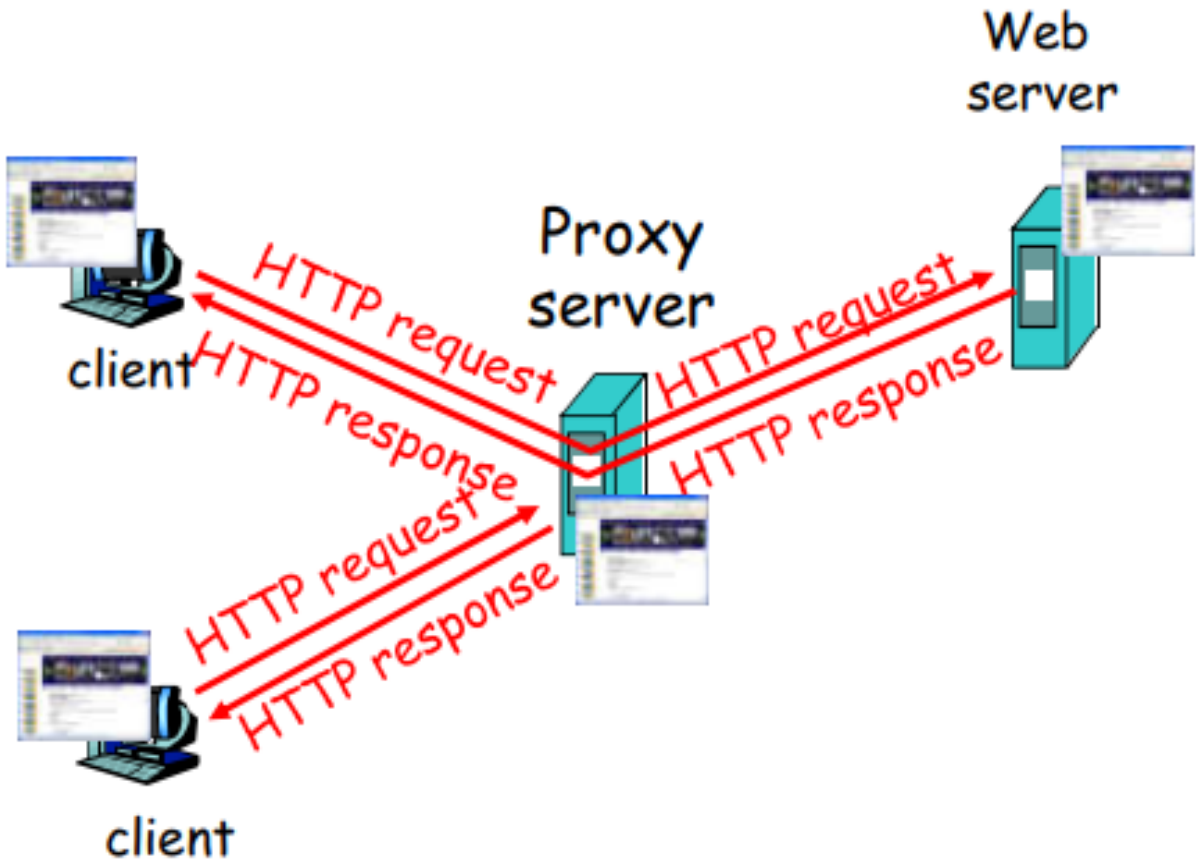
Bộ đệm- Caching

- “Cache”: Bộ nhớ đệm
- Khái niệm bộ nhớ cache trong máy tính
- L1 cache, L2 cache
- “cache miss”, “cache hit”
- Xem xét trường hợp sau:
- Một tổ chức có một đường nối tới Internet
- Tất cả lưu lượng truy cập web đều đi qua liên kết này
- Nhiều NSD web có thể cùng truy nhập tới cùng một nội dung
- Giải pháp cải tiến?



Sử dụng bộ đệm - web proxy

NSD đặt tham số kết nối

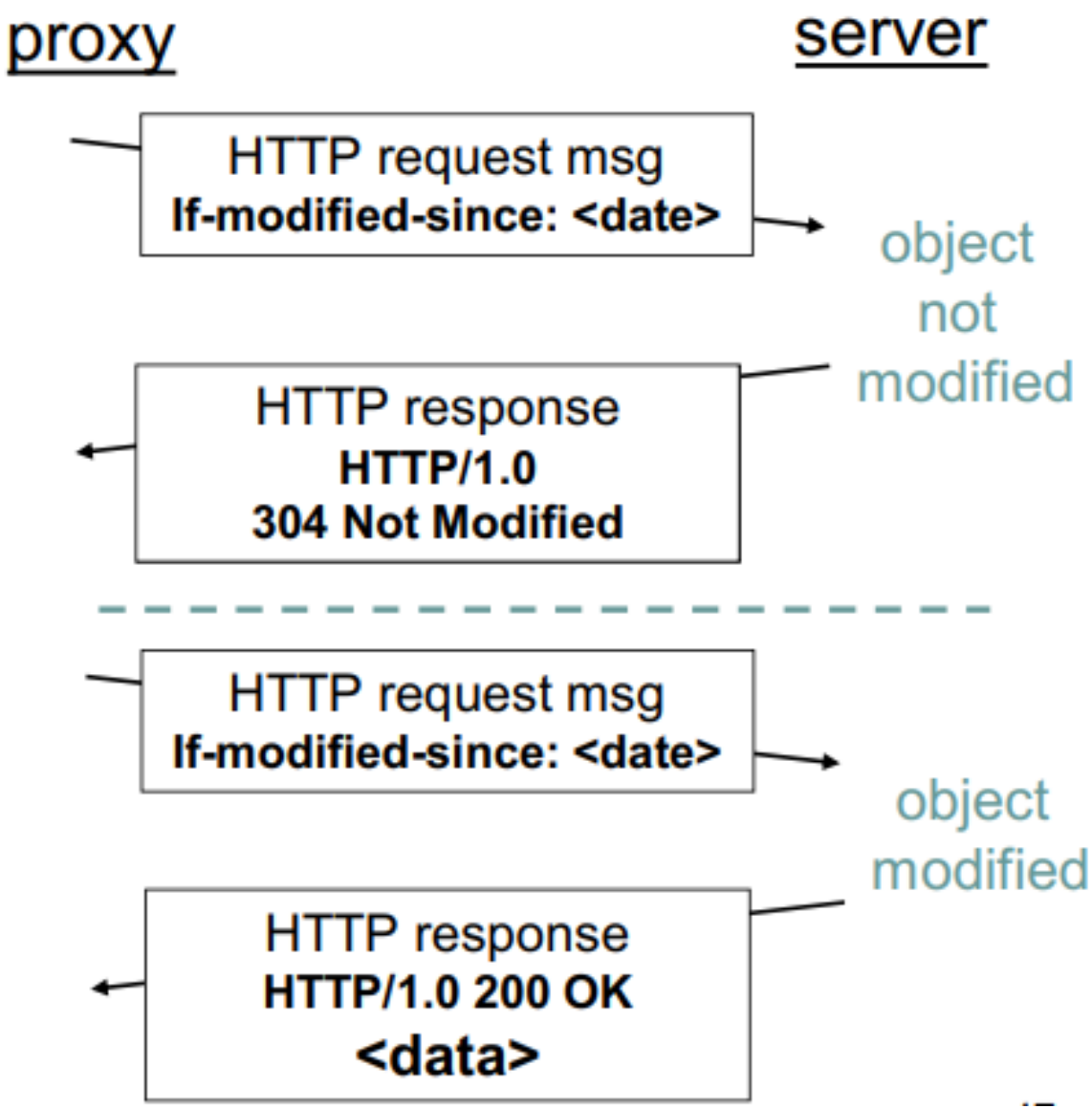


truy cập web của trình duyệt qua một máy chủ proxy

- trình duyệt gửi yêu cầu đến proxy
- Miss: Proxy gửi yêu cầu tới máy chủ web, trả lời trình duyệt và lưu đệm đối tượng web
- Hit: Proxy trả đối tượng web cho trình duyệt

Phương thức GET có điều kiện

- Mục đích: Máy chủ sẽ không gửi đối tượng web nếu proxy còn lưu giữ thông tin cập nhật
- Proxy: chỉ ra thời gian cũ của đối tượng
If-modified-since: <date>
server: Xác nhận lại có thay đổi hay không:
HTTP/1.0 304 Not Modified



Web proxy

- Proxy: Vừa là client, vừa là server
- Sử dụng bởi các ISP nhỏ, các tổ chức như trường học, công ty…
- Ảnh hưởng của proxy
- Làm giảm lưu lượng web trên đường ra Internet
- Có thể làm giảm thời gian đáp ứng
- Thử phân tích vài trường hợp
- cache hit
- cache miss
- proxy bị quá tải
- Trang web thay đổi/trang web động?

Local cache

- Các trang web còn có thể được lưu trên máy cục bộ
- Sử dụng local cache để
  - Duyệt web offline
  - Duyệt các trang web hiệu quả hơn

HTTPS

- Hạn chế của HTTP:
- Không có cơ chế để người dùng kiểm tra tính tin cậy của Web server → lỗ hổng để kẻ tấn công giả mạo dịch vụ hoặc chèn mã độc vào trang web HTML
- Không có cơ chế mã hóa giữ mật → lỗ hổng để kẻ tấn công nghe lén đánh cắp thông tin nhạy cảm
- Secure HTTP: sử dụng liên kết SSL/TLS thay cho TCP để truyền các thông điệp HTTP
- Xác thực:
- Người dùng truy cập vào đúng Website mong muốn
- Dữ liệu trong quá trình truyền không bị thay đổi
- Bảo mật: dữ liệu được giữ bí mật trong quá trình truyền
- Số hiệu cổng ứng dụng: 443

HTTPS

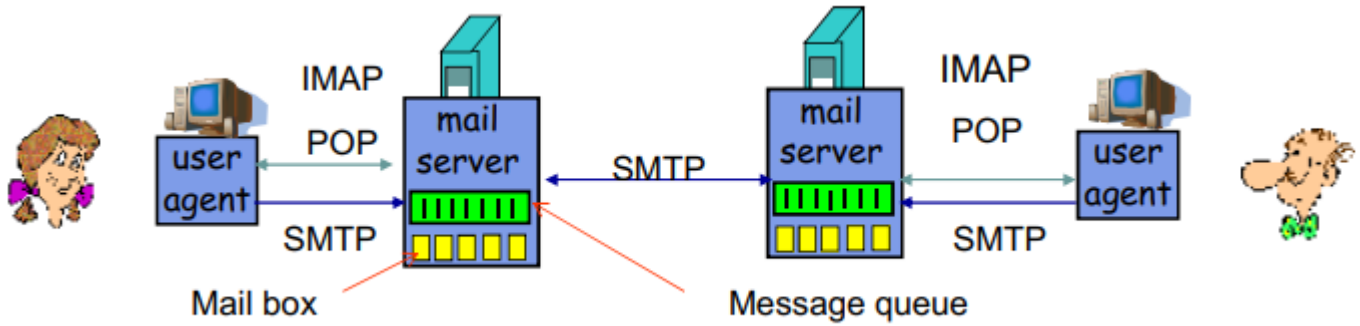
- Tuy nhiên, HTTPS có thể gây hiểu nhầm cho người dùng rằng trang web là an toàn:
- Người dùng bất cẩn vì chỉ chú ý biểu tượng

- HTTPS không chống lại được các dạng tấn công khai thác điểm yếu của Website

Thư điện tử

- MUA (Mail User Agent)
  - Lấy thư từ máy chủ, gửi thư đến máy chủ
  - e.g. Outlook, Thunderbird…
- MTA (Mail Transfer Agent): :
  - Chứa hộp thư đến của NSD (mail box)
  - Hàng đợi để gửi thư đi
  - e.g. Sendmail, MSExchange…

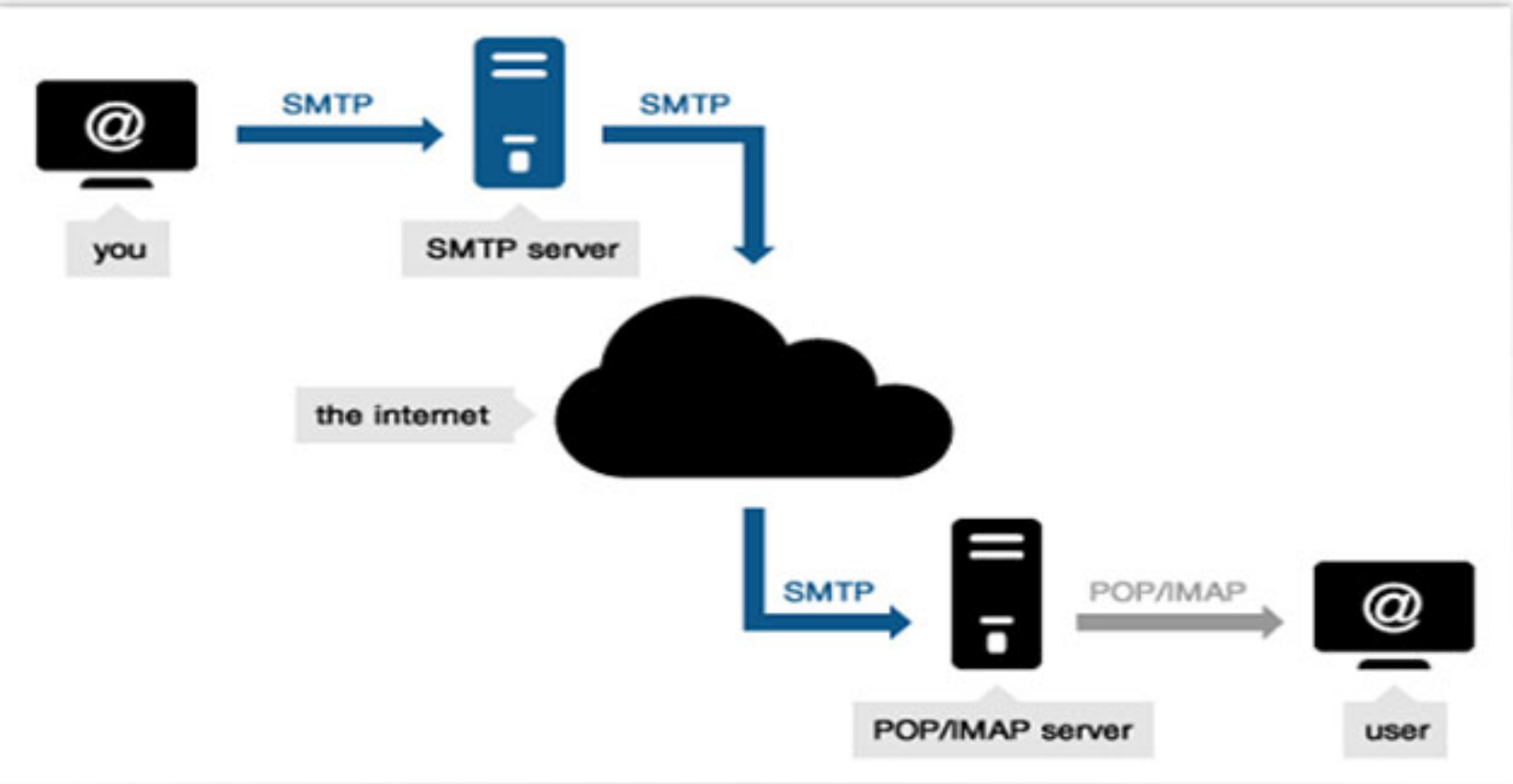
- Giao thức:
  - Chuyển thư: SMTP-Simple Mail Transfer Protocol
  - nhận thư
  - POP – Post Office Protocol
  - IMAP – Internet Mail Access Protocol



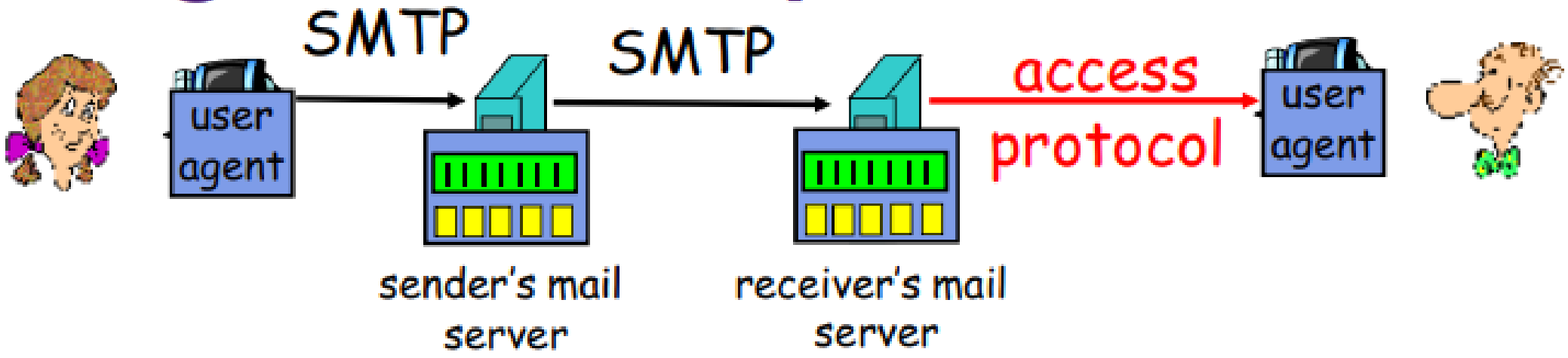
Giao thức SMTP

- RFC 2821
- TCP, port 25: Chuyển thư từ client đến server và giữa các server với nhau
- Tương tác yêu cầu/trả lời
- Yêu cầu: Lệnh với mã ASCII
- Trả lời: mã trạng thái và dữ liệu

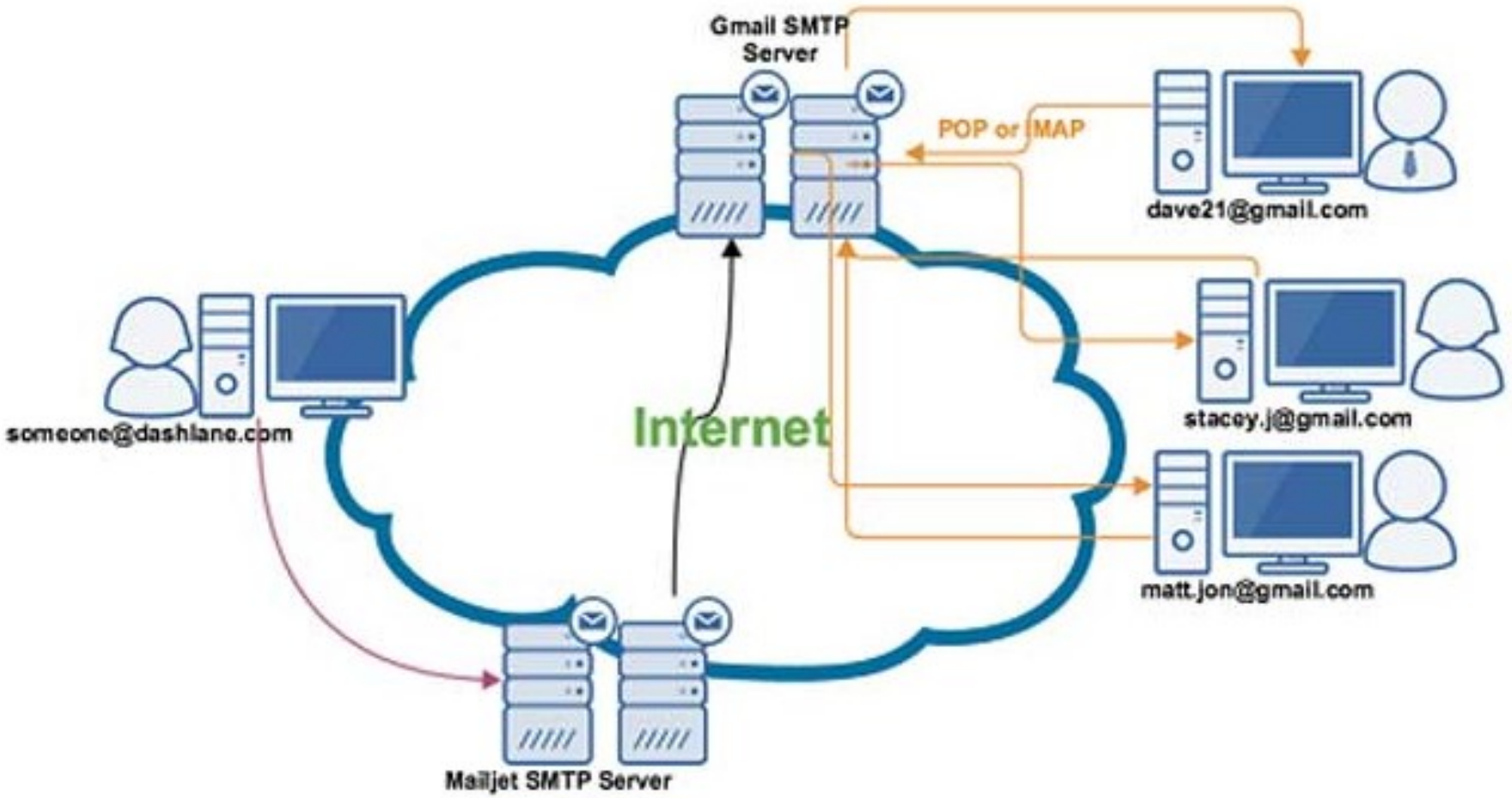
Giao thức SMTP



Các giao thức nhận thư



- POP: Post Office Protocol [RFC 1939]
- Đăng nhập và lấy hết thư về
- IMAP: Internet Mail Access Protocol [RFC 1730]
- Phức tạp hơn POP
- Cho phép lưu trữ và xử lý thư trên máy chủ



Cách thức hoạt động của hệ thống SMTP

Web Mail

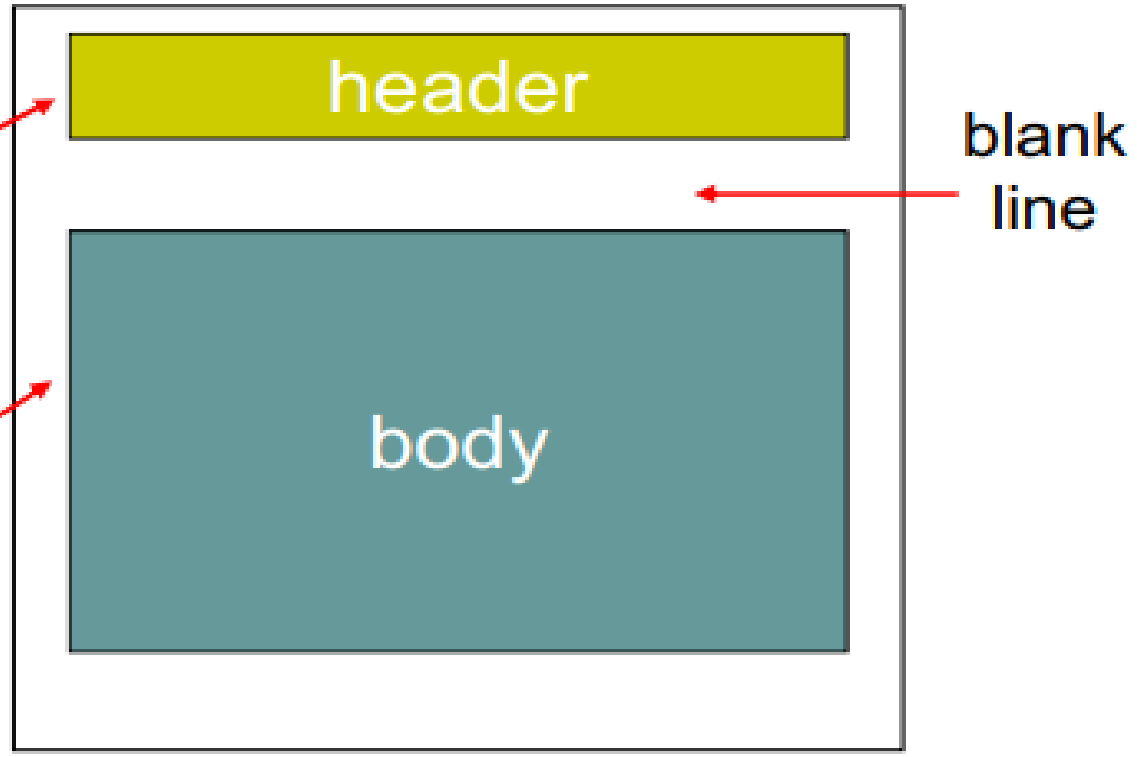
- Sử dụng Web browser như một MUA
- MUA và MTA giao tiếp thông qua HTTP
- Mails được lưu trữ trên máy chủ
- E.g.
  - Gmail,
  - Hotmail,
  - Yahoo! Mail, etc.
- Ngày nay, rất nhiều các MTA cho phép truy cập thông qua giao diện web
  - http://mail.iuh.edu.vn
  - http://mail.FET.iuh.edu.vn

Khuôn dạng thông điệp thư điện tử

SMTP: Giao thức để truyền thư

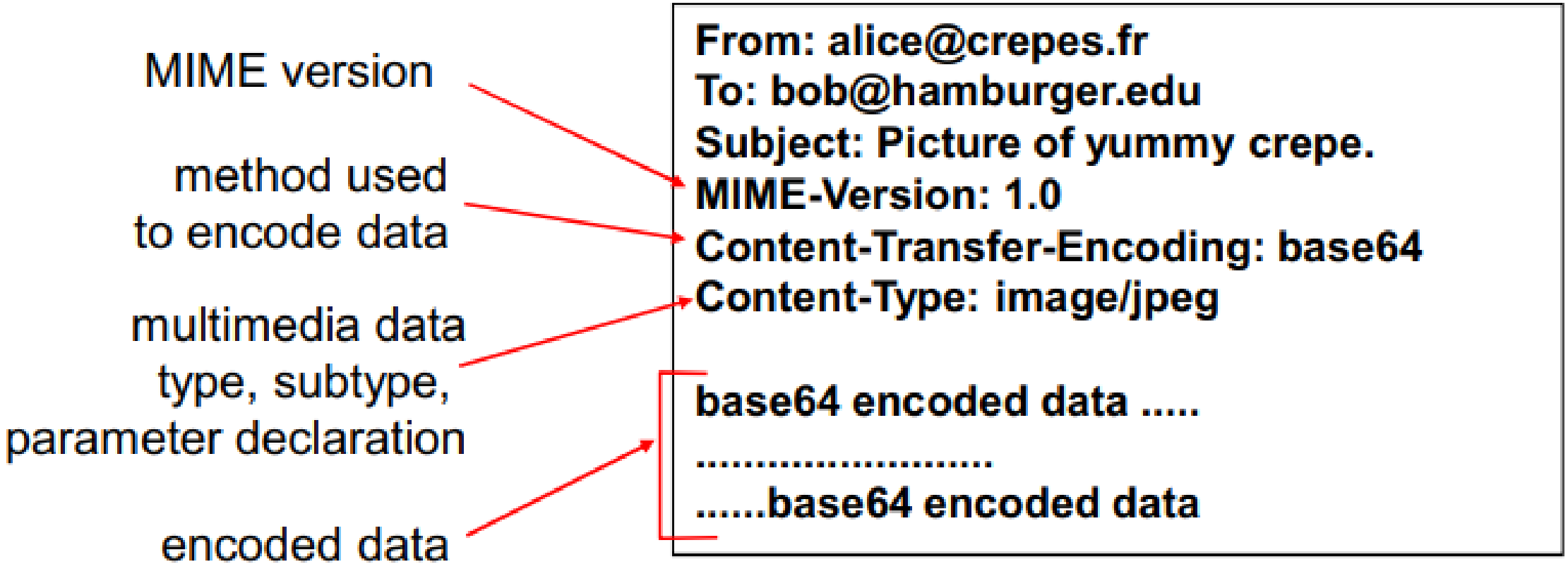
RFC 822: Định nghĩa khuôn dạng

- Phần đầu
  - To:
  - From:
  - Subject:
- Phần thân
  - mã hóa dưới dạng mã ASCII

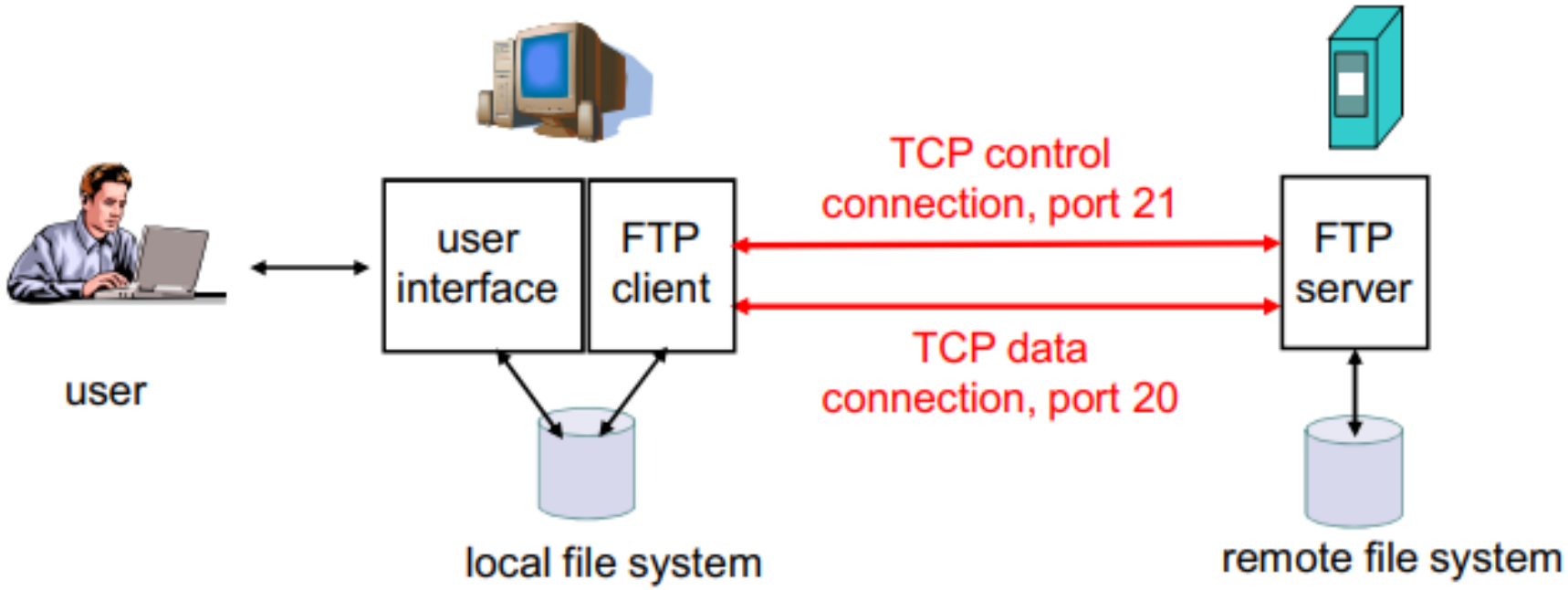


Để chuyển dữ liệu đa phương tiện: multimedia extensions

- MIME: multimedia mail extension, RFC 2045, 2056
- Thêm một dòng trong phần đầu chỉ rõ khuôn dạng dữ liệu gửi đi



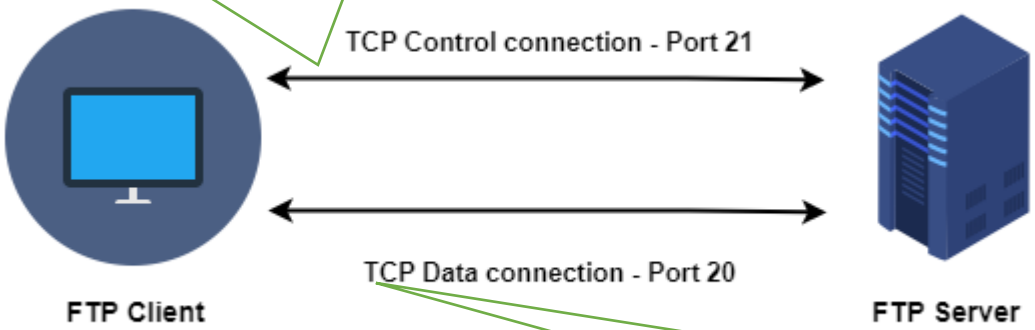
FTP: File Transfer Protocol



- Mô hình Client-server
- Trao đổi file giữa các máy
- RFC 959
- Sử dụng TCP, cổng 20, 21

- Điều khiển Out-of-band :
- Lệnh của FTP : cổng 21
- Dữ liệu: cổng 20
- NSD phải đăng nhập trước khi truyền file
- Một số server cho phép NSD với tên là anonymous

Control connection (sử dụng port 21 – trên server): Đây là kết nối TCP logic chính được tạo ra khi phiên làm việc được thiết lập. Nó được thực hiện giữa các quá trình điều khiển. Nó được duy trì trong suốt phiên làm việc và chỉ cho các thông tin điều khiển đi qua như lệnh hay response(phản hồi)



Data connection (sử dụng port 20 – trên server): Kết nối này sử dụng các quy tắc rất phức tạp vì các loại dữ liệu có thể khác nhau. Nó được thực hiện giữa các quá trình truyền dữ liệu. Kết nối này mở khi có lệnh chuyển tệp và đóng khi tệp truyền xong.

FTP chia mỗi thiết bị thành 2 phần giao thức logic chịu trách nhiệm cho mỗi kết nối ở trên:

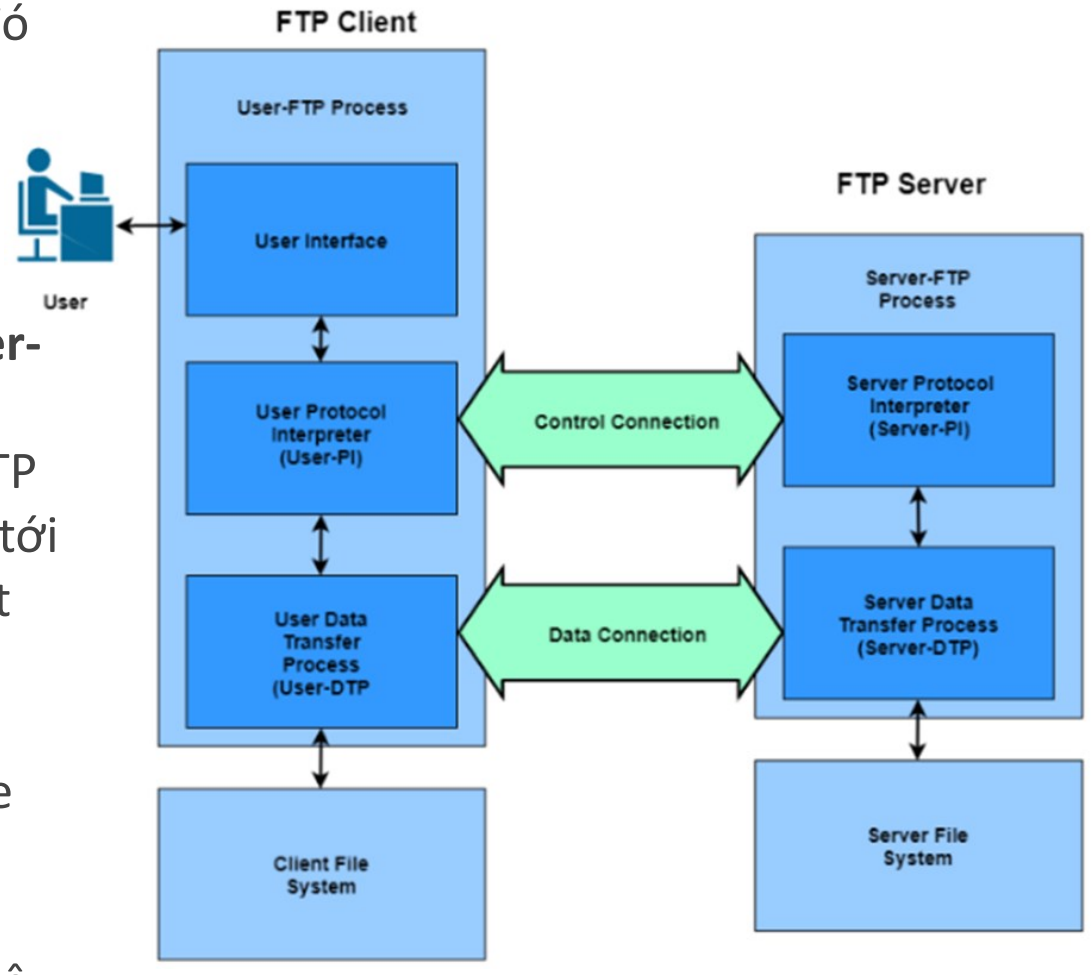
- **Protocol interpreter (PI)**: Là thành phần quản lý kênh điều khiển, phát và nhận lệnh và trả lời.

- **Data transfer process (DTP)**: chịu trách nhiệm gửi và nhận dữ liệu giữa client và server.

•**User Interface**: user cung cấp giao diện xử lí cho người dùng, để điều khiển các session FTP, từ đó có thể theo dõi được các thông tin và kết quả xảy ra trong quá trình.

•**User Protocol Interpreter (User-PI)**: quản lí Control Connection phía Client. Tạo phiên kết nối FTP bằng việc phát hiện ra Request tới Server-PI. Khi kết nối được thiết lập, xử lí lệnh nhận được trên User Interface, gửi chúng tới Server-PI rồi đợi nhận Response trở lại.

•**User Data Transfer Process (User-DTP)**: gửi hoặc nhận dữ liệu từ Server-DTP. User-DTP thiết lập hoặc lắng nghe DataConnection từ Server thông qua cổng 20.



•**Server Protocol Interpreter (Server-PI)** : Chịu trách nhiệm quản lí Control Connection trên Server. Lắng nghe yêu cầu kết nối hướng từ User trên cổng 21. Khi kết nối được thiết lập, nó nhận lệnh từ User-PI, gửi phản hồi và quản lí tiến trình truyền dữ liệu trên Server.

•**Server Data Transfer Process (Server-DTP)** : nhận và gửi file từ User-DTP. Server-DTP làm nhiệm vụ thiết lập Data Connection và lắng nghe Data Connection của User thông qua cổng 20. Nó tương tác với Server File System trên hệ thống cục bộ để đọc và chép file.

Ví dụ về ftp client

Command line

C:\Documents and Settings\hongson>ftp
ftp> ?
Commands may be abbreviated. Commands are:

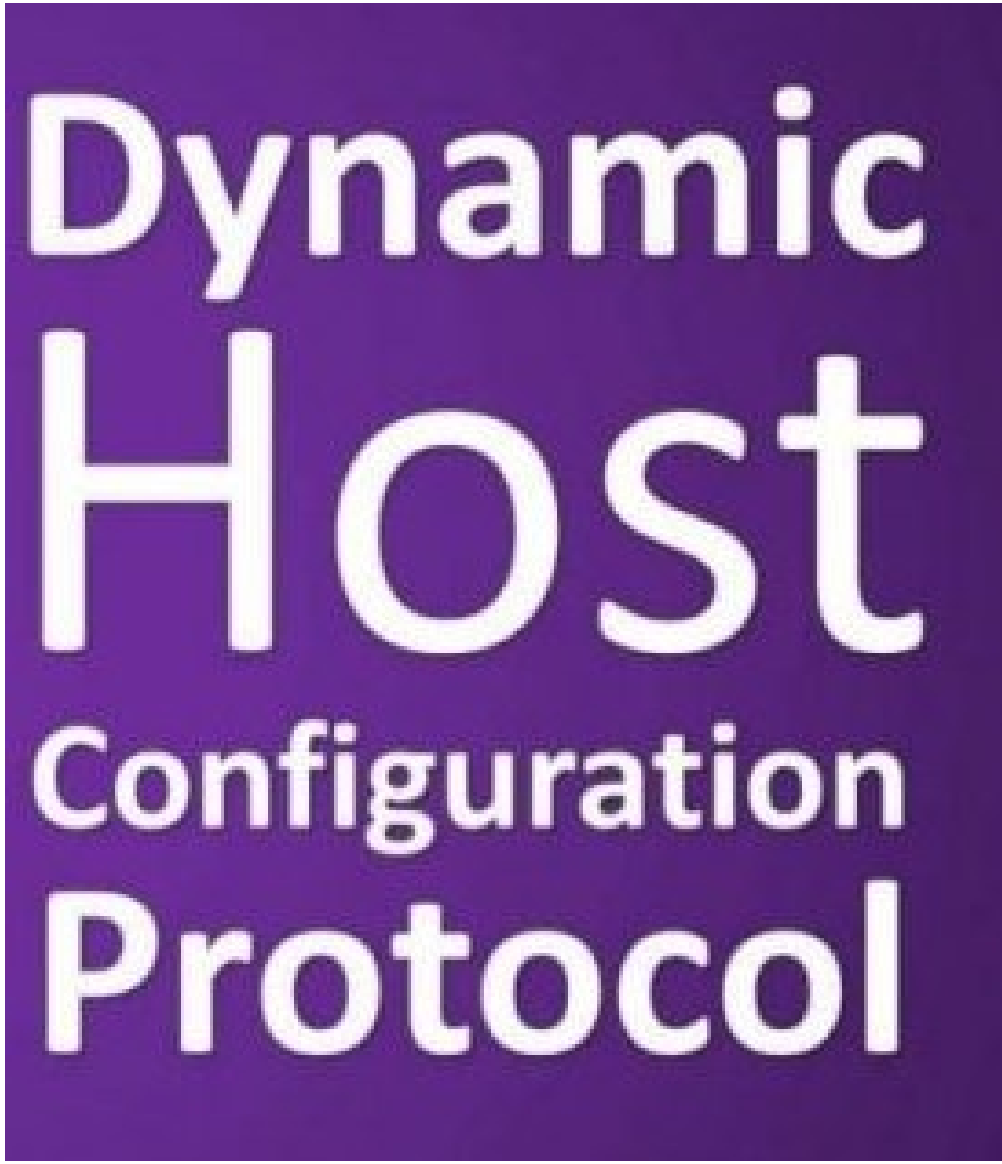
!	**delete**	literal	prompt	send
?	debug	ls	**put**	status
append	dir	mdelete	pwd	trace
ascii	disconnect	mdir	quit	type
bell	**get**	mget	quote	user
binary	glob	mkdir	recv	verbose
bye	hash	mls	remotehelp	
cd	help	mput	rename	
close	lcd	**open**	rmdir	

GUI FTP clients: IE, Firefox, GFTP, ….

DHCP

- DHCP là giao thức được sử dụng để cấu hình địa chỉ IP cho thiết bị máy tính.

- Khi máy tính lần đầu tiên kết nối tới mạng nội bộ bằng dây cáp hoặc truy cập Wifi, điều đầu tiên nó làm chính là tìm địa chỉ IP, netmask, default gateway và DNS servers.



DHCP server có thể có 3 phương thức cấp phát địa chỉ IP:

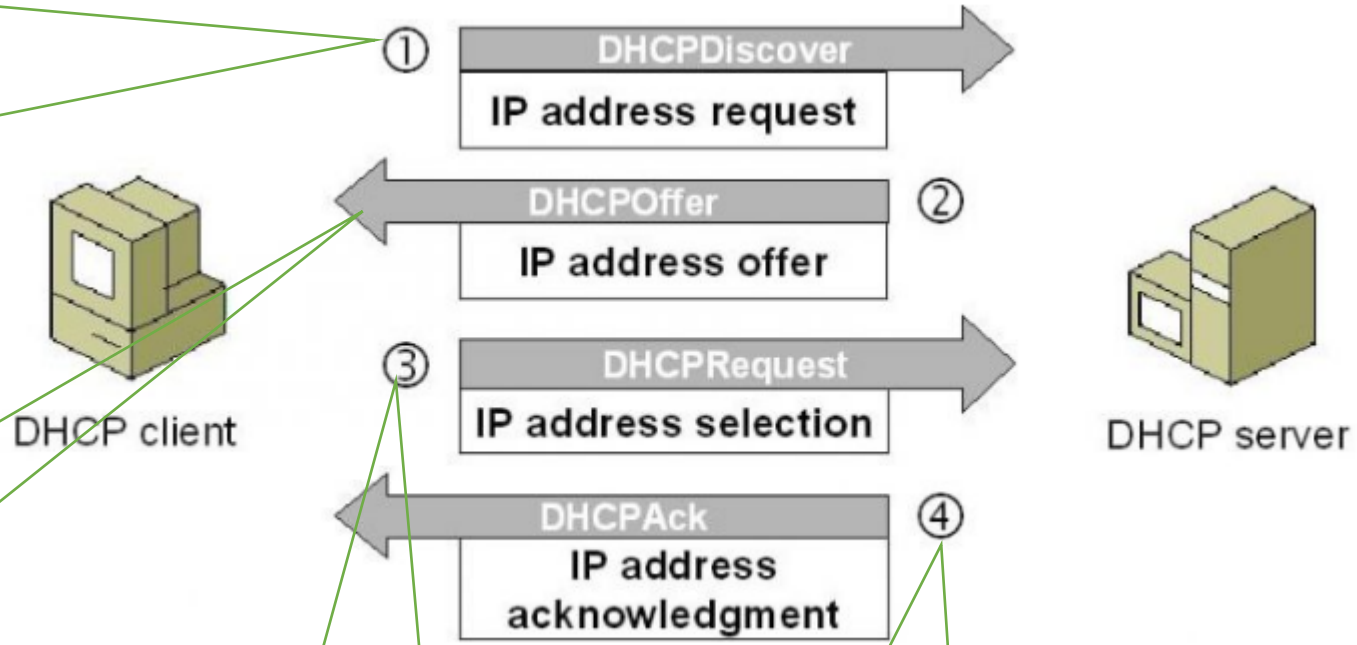
- **Cấp phát tĩnh:** DHCP server cấp phát một địa chỉ IP dựa trên một bảng với cặp địa chỉ MAC/ địa chỉ IP tương ứng, được điền thủ công và chỉ khi có yêu cầu từ client với địa chỉ MAC được liệt kê bên trong bảng mới được cấp IP.

- **Cấp phát động:** người quản trị mạng sẽ gán một dãy (range) địa chỉ IP tới DHCP, và mỗi máy tính client trong mạng LAN được cấu hình để request một địa chỉ IP từ DHCP server trong quá trình khởi tạo mạng.

- **Cấp phát tự động:** DHCP server gán vĩnh viễn địa chỉ IP tới request client từ dãy địa chỉ IP được quy định bởi người quản trị. Tương tự như cấp phát động, nhưng DHCP server giữ lại bảng chứa các địa chỉ IP được gán trước đó, để nó có thể gán cho client cùng địa chỉ IP mà họ đã request trước đó.

DHCP cung cấp tự động. Một DHCP server cung cấp thông tin này tới DHCP client thông qua 4 bước.

Bước 1: DHCP Discovery
Máy tính client sẽ gửi thông điệp broadcast trên physical subnet để tìm server DHCP khả dụng. Máy tính client tạo ra một gói tin UDP đích đến mặc định 255.255.255.255 hoặc địa chỉ broadcast subnet cụ thể nếu được cấu hình.



Bước 2: DHCP offer
DHCP server nhận được y/c cấp IP từ client, bảo lưu địa chỉ IP cho client và mở rộng địa chỉ IP sẽ gửi cho client message DHCPOFFER. Thông điệp này chứa địa chỉ MAC của client, địa chỉ IP mà server sẽ cung cấp, subnet mask,

Bước 3: DHCP request
Client trả lời DHCP request, gửi tin unicast tới server địa chỉ nằm trong DHCP offser nhận được.

Bước 4: DHCP acknowledgement
Khi DHCP server nhận được thông điệp DHCPREQUEST từ client, quá trình cấu bước vào giai đoạn cuối cùng.

Telnet

Telnet được viết tắt bởi các cụm từ **“teletype network“**, **“terminal network”** hay **“telecommunications network“.**

Cấu trúc và nguyên lý hoạt động của Telnet

- Cấu trúc đơn giản với mô hình Client-Server. Trong đó gồm Client và Server. Phía Server sẽ cung cấp các dịch vụ Telnet để kết nối ứng dụng Telnet của máy Client.

- Máy Client phải xác định rõ cổng Telnet vì phía máy chủ Telnet sẽ lắng nghe cổng TCP 23 để xác định Telnet.

- Người dùng kết nối từ xa với một máy bằng Telnet (việc này được gọi là Telneting). Có thể sử dụng các dòng lệnh trực tiếp bằng máy tính từ xa sau khi đăng nhập. Địa chỉ IP sẽ luôn khớp với máy tính đó bất kể vị trí địa lý của người dùng.

Lệnh cơ bản khi sử dụng Telnet

- cd: Dùng để đổi từ thư mục này sang thư mục khác
- Pwd: Cho biết vị trí hiện tại của hệ thống, ví dụ như bạn đang ở thư mục nào
- Is-a: Lệnh liệt kê tất cả các file, ngay cả các file bị ẩn
- Is-l: Lệnh liệt kê các file chi tiết
- Is-la: Lệnh liệt kê các file thông dụng